Image and Video Analysis for Intelligent Driver Monitoring in Car Cabins

Mathias Viborg Andersen

Computer Engineering, CE10, May 31, 2024

AALBORG
UNIVERSITET

Electronic Systems
Fredrik Bajers Vej 7B
9220 Aalborg Øst

Title:

Image and Video Analysis for Intelligent Driver Monitoring in Car Cabins

Theme:

Masters Thesis: Computer Engineering: AI, Vision and Sound

Project:

P10 - Master Project

Project Period:

Spring 2024

Project Group:

CE AVS 1045a

Participant:

Mathias Viborg Andersen

Supervisor:

Andreas Møgelmose, Aalborg University

Co-supervisors:

Ross Greer, UC San Diego
Mohan M. Trivdei, UC San Diego

Pages: 83
Date of Completion: May 31, 2024

Abstract:

This work consists a master's thesis conducted in Computer Engineering: AI, Vision & Sound. It describes the work conducted during a semester abroad at UC San Diego. It is research-oriented and is therefore structured in separate parts to align with the flow of research in contrast to a linear development flow.

The primary scope of this study has been the exploration of synthetic generation of missing thermal video frames, generalizable driver activity classification and driver drowsiness detection, all within the context of a car cabin. This work includes a proposal for generating missing thermal frames from RGB, achieving well-performing results through the use of conditional Generative Adversarial Networks (cGANs). Additionally, an experiment utilizing multiple camera angles in Vision-Language models for in-car-cabin activities have been conducted, showcasing promising results for generalizable Vision-Language models. Furthermore, a study utilizing Video Transformers in the pursuit of drowsiness classification has been conducted detailing the accuracy of video detail needed for the task, and includes a custom facial video cropped version of the UTA-RLDD.

This work has resulted in the acceptance of a paper for the 35th IEEE Intelligent Vehicles Symposium (IV) as well as an acceptance at the Computer Vision and Pattern Recognizion (CVPR) Vision and Language for Autonomous Driving and Robotics Workshop.

Elektroniske Systemer
Fredrik Bajers Vej 7B
9220 Aalborg Øst

Titel:

Billede- og Videoanalyse til Intelligent Førerassistance i Bilkabiner

Tema:

Kandidat Projekt: Computer Engineering: AI, Vision and Sound

Projekt:

P10 - Kandidat Speciale

Projekt Periode:

Forår 2024

Projekt Gruppe:

CE AVS 1045a

Deltager:

Mathias Viborg Andersen

Vejleder:

Andreas Møgelmose, Aalborg Universitet

Medvejledere:

Ross Greer, UC San Diego
Mohan M. Trivdei, UC San Diego

Sider: 83

Afleveringsdato: 31. Maj 2024

Abstract:

Dette arbejde består af et kandidatspeciale i Computer Engineering: AI, Vision & Sound. Det beskriver det arbejde, der er udført under et semesterophold ved UC San Diego. Det er forskningsorienteret og struktureret i separate dele for at tilpasse sig forskningsflowet i modsætning til en lineær udviklingsproces.

Det primære omfang af denne rapport har været undersøgelsen af syntetisk generering af manglende termiske video-frames, generaliserbar klassificering af chaufføraktivitet og registrering af førertræthed, alt sammen inden for konteksten af en bilkabine. Denne rapport inkluderer et forslag til generering af manglende termiske rammer fra RGB, hvor der opnås brugbare resultater ved brug af conditional Generative Adversarial Networks (cGANs). Derudover er der udført et eksperiment med anvendelse af flere kameravinkler i Vision-Language modeller til aktiviteter i bilkabinen, som viser lovende resultater for generaliserbare Vision-Language modeller. Ydermere er der udført en undersøgelse, der benytter Video Transformers i forsøget på at klassificere træthed, som detaljerer den nødvendige videodetalje for opgaven, og inkluderer en ansigtstilpasset version af datasættet UTA-RLDD.

Dette arbejde har resulteret i en accepteret artikel til det 35. IEEE Intelligent Vehicles Symposium (IV) samt en accept ved Computer Vision and Pattern Recognition (CVPR) Vision and Language for Autonomous Driving and Robotics Workshop.

Preface

Mathias Viborg Andersen authored this report during the period spanning from February 2024 to May 2024. The thesis represents the culmination of the Master's degree in Computer Engineering, specializing in *Artificial Intelligence, Vision and Sound* at Aalborg University, Denmark. The primary research was conducted in collaboration with the Computer Vision and Robotics Research (CVRR) Laboratory and the Laboratory for Intelligent and Safe Automobiles (LISA) at the University of California, San Diego (UCSD), CA, USA, with significant contributions made abroad.

The utilized code for Chapter 2 & 4 can be found at https://github.com/viborgen/p10. The utilized code for Chapter 3 can be found at https://github.com/viborgen/Driver-Activity-Classification-Using-Generalizable-Representations-from-Vision-Language-Models.

Acknowledgement

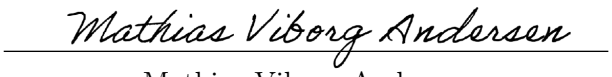
Sincere gratitude is extended to Professor Mohan M. Trivedi for providing the opportunity to experience the workflow in a laboratory setting and see the beautiful city of San Diego, and to Ross Greer for his always helpful attitude, supervision, long discussions, constant question answering and guidance throughout the duration of this project.

Furthermore, a special thanks is extended to Andreas Møgelmose for facilitating the collaboration that made this project possible, the members of LISA and CVRR who warmly welcomed me, making me feel included and part of the lab community right from the start, and my wonderful roommates for being fun to live with and enriching the experience.

Special thanks are also extended to Reza Ghoddoosian from Honda Research Institute USA for kindly providing the complete UTA-RLDD dataset via email.

Reading

All figures, tables and equations are named sequentially according to the chapter they are in. The figures in Chapter 2 will be *Figure 2.1*, *Figure 2.2* and so forth. The decimal separator is the decimal point.



Mathias Viborg Andersen
mvan19@student.aau.dk
mathias.v.andersen@gmail.com

Table of Contents

Introduction 1

This thesis is the result of my stay at LISA and CVRR at UC San Diego spanning three main work areas all in relation to intelligent vehicles. Research has been conducted on thermal video frame generation, classification using generalizable representations from Vision-Language models, and drowsiness detection. These studies are detailed in Chapter 2, Chapter 3, and Chapter 4, respectively. The work has resulted in two scientific papers, both accepted for publication. Both papers are attached in Appendix D.

- Learning to Find Missing Video Frames with Synthetic Data Augmentation: A General Framework and Application in Generating Thermal Images Using RGB Cameras. Accepted at the 35th IEEE Intelligent Vehicles Symposium (IV).
- Driver Activity Classification Using Generalizable Representations from Vision-Language Models. Accepted at the Computer Vision and Pattern Recognizion (CVPR) Vision and Language for Autonomous Driving and Robotics Workshop.

It is expected that the paper *Driver Activity Classification Using Generalizable Representations from Vision-Language Models* will also be submitted elsewhere and may take another form after the submission of this report, as the CVPR workshop is non-archival.

The work has been carried out with the supervision of PhD candidate Ross Greer from UCSD and associate professor Andreas Møgelmose from Aalborg University. Chapter 3 has been carried out in collaboration with Ross Greer, with the main contributions being done by me.

A common theme across all chapters is that they focus on activities within the vehicle cabin, *looking in*, and are centered on human interaction, though in different specific areas. The report does not have a linear flow as a result, with each chapter focusing on independent areas. Below is a brief overview of the chapters:

1.1 Chapter 2: Learning to Find Missing Video Frames with Synthetic Data Augmentation

In Chapter 2, the generation of thermal frames from corresponding RGB frames is explored to address the issue of missing data due to sensor frame rate mismatches. By leveraging conditional Generative Adversarial Networks (cGANs), specifically comparing the pix2pix and CycleGAN architectures, this chapter demonstrates the potential of generative models in creating realistic thermal images. The experimental results indicate that pix2pix outperforms CycleGAN and that using multi-view input styles, particularly stacked views, enhances the accuracy of thermal image

generation. This work contributes to advancing driver state monitoring systems by providing a method to maintain data integrity when frame rates diverge.

1.2 Chapter 3: Driver Activity Classification Using Generalizable Representations from Vision-Language Models

Chapter 3 presents a novel approach for driver activity classification using Vision-Language models. This research leverages generalizable representations to classify in-cabin activities such as drinking, talking on the phone, and texting, without the need for extensive model fine-tuning. The study applies theoretical concepts to practical scenarios, including the 2024 AI City Challenge, specifically Track 3: Naturalistic Driving Action Recognition. This chapter underscores the flexibility of Vision-Language models and their applicability to a broad range of tasks, showcasing their potential for robust driver monitoring systems.

1.3 Chapter 4: Drowsiness Detection Utilizing Video Transformers

Chapter 4 delves into the detection of driver drowsiness using Video Transformers, specifically the TimeSformer model. This chapter discusses the indicators of drowsiness and related works, followed by a temporal analysis strategy employing TimeSformer. The research demonstrates how TimeSformer can effectively process and analyze video data to identify drowsiness, highlighting its competitive performance in action recognition tasks.

1.4 Utilized hardware

Throughout the development of the project, three primary hardware configurations were employed: two desktops, *LISA Desktop 1* and *LISA Desktop 2*, and a cloud instance, *AAU Strato Cloud Instance*. As projects often have been running simultaneously each chapter has had a main place of operation. Chapter 2 has mainly utilized *LISA desktop 1*, Chapter 3 has mainly utilized *LISA desktop 2*, and Chapter 4 has mainly utilized *AAU Strato Cloud Instance*.

LISA desktop 1:

- CPU: AMD Ryzen 9 5950X 16-Core Processor.
- GPU: NVIDIA GeForce RTX 3090.

LISA desktop 2:

- CPU: AMD Ryzen 9 7950X 16-Core Processor.
- GPU: NVIDIA GeForce RTX 4090.

AAU Strato Cloud Instance:

- CPU: Intel Xeon Processor (Icelake).
- GPU: NVIDIA A10.

Learning to Find Missing Video Frames with Synthetic Data Augmentation

2

This work has been accepted for the 35th IEEE Intelligent Vehicles Symposium (IV), scheduled for June 2 to June 5, 2024. An article of the name *Learning to Find Missing Video Frames with Synthetic Data Augmentation: A General Framework and Application in Generating Thermal Images Using RGB Cameras* can be found in Appendix D. It is advisable to read the article before engaging with this chapter, as this chapter build upon the article's understanding.

This chapter aims to explore the generation of thermal frames from a corresponding RGB frame given a sensor frame rate mismatch. Building on the insights from the attached article and Section 4.3, where the challenge of dealing with varying frame rates in datasets is detailed (as depicted in Figure 4.3a), the significance of maintaining data integrity when frame rates diverge is underscored.

2.1 Generative strategy

When exploring the generation of non-existent thermal data frames from corresponding existing RGB frames that capture the same elements, conditional Generative Adversarial Networks (cGANs) present a promising approach [1]. These networks extend the standard Generative Adversarial Network (GAN) architecture [2] by incorporating a conditional element, allowing for targeted generation of images based on given inputs. Unlike traditional GANs, which solely rely on a generator and discriminator competing in a minimax game, cGANs introduce an additional condition, typically in the form of a class label or related data, to guide the image generation process more precisely as on Figure 2.1.

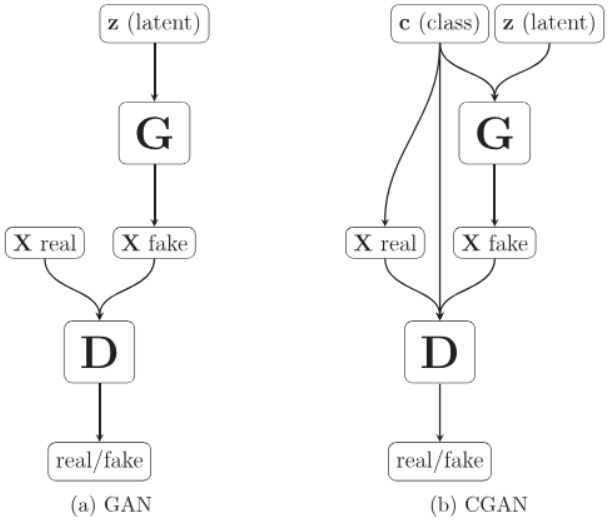


Figure 2.1. GAN and cGAN compared. G is generator, D is discriminator, X represents samples from dataset and generated samples, c the added class or label [3].

Two notable examples of cGAN architectures tailored for image-to-image translation tasks are pix2pix [4] and CycleGAN [5]. Both will be further explained in this section.

2.1.1 pix2pix

The pix2pix framework is designed for tasks where paired input-output examples are available, facilitating direct translations from one domain to another. This approach is particularly useful for applications requiring precise alignment between the input and generated images, as it leverages the explicit correspondence between the pairs, as in Figure 2.2.

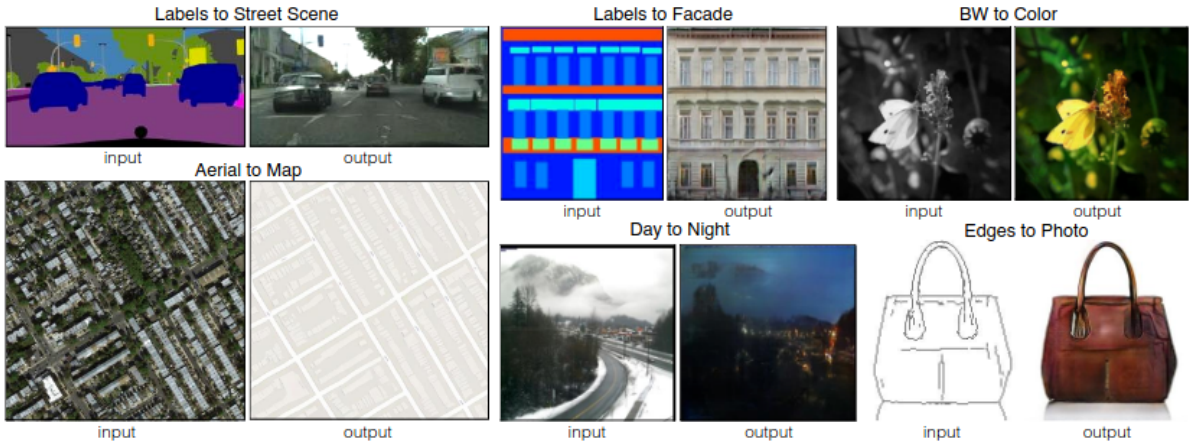


Figure 2.2. Example from the pix2pix architecture directly translating from one modality to another, depending on paired inputs and outputs [4].

The generator (G) in pix2pix aims to produce output images that are indistinguishable from real images in the target domain. Unlike conventional GANs, the pix2pix generator takes an image from the source domain as input and generates a corresponding image in the target domain, instead of using random noise Z. The generator architecture is based on a U-Net structure,

which features a series of down-sampling layers followed by up-sampling layers, connected with residual connections. The U-Net architecture enables the network to capture context from the input image at different levels of detail and efficiently reconstruct the output image with high fidelity. The function of the discriminator (D) is to differentiate between real images from the target domain and fake images produced by the generator. The pix2pix discriminator operates on patches of the image, a design known as PatchGAN. This design allows the discriminator to focus on high-frequency details by classifying each patch as real or fake, making it more effective in assessing the authenticity of local image textures and structures compared to looking at individual pixels (PixelGAN) or full images (ImageGAN).

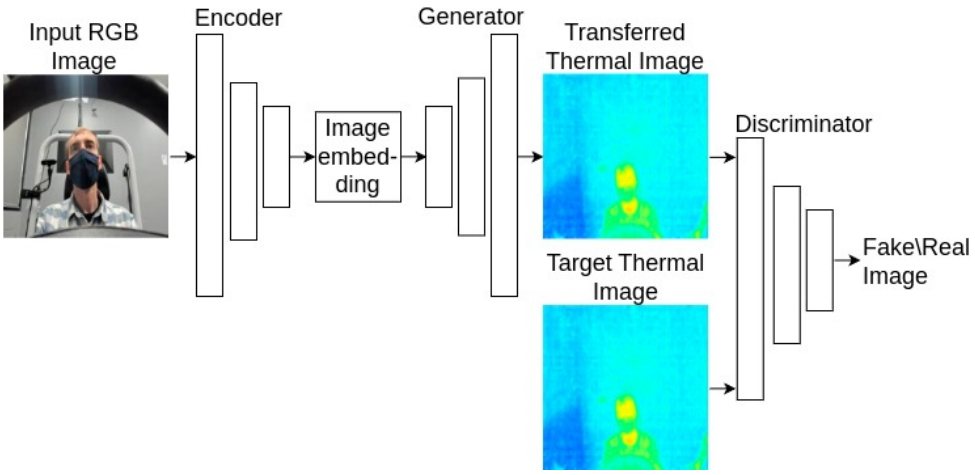


Figure 2.3. The flow of pix2pix applied in this work.

The pix2pix training process involves a combination of two main loss functions:

Adversarial Loss: This loss measures how well the discriminator is able to distinguish between real and generated images. The generator is trained to minimize this loss by trying to produce images that the discriminator will classify as real. This creates a competitive game between the generator and discriminator, driving the generator to produce increasingly realistic images. In this implementation, binary cross-entropy (BCE) loss is used [6].

L1 Loss: To ensure that the generated images not only fool the discriminator but also are close to the real images in a meaningful way, pix2pix includes an L1 loss (also known as the mean absolute error) between the generated image and the real target image. This loss encourages the generated image to be similar to the target image on a pixel-by-pixel basis, thus preserving the content of the input image in the generated output. The authors argue that it encourages less blurring than L2 loss.

These two losses are collected in equation 2.1, which is the final objective of the pix2pix cGAN.

$$\mathrm{G}^* = \arg\min_{\mathrm{G}} \max_{\mathrm{D}} \mathcal{L}_{\mathrm{cGAN}}(\mathrm{G}, \mathrm{D}) + \lambda \mathcal{L}_1(\mathrm{G}) \tag{2.1}$$

G^*	The generator model that minimizes the combined adversarial and L1 losses to produce images closest to the target distribution.
$\mathcal{L}_{\text{cGAN}}(G, D)$	The adversarial loss calculated from both the generator's and the discriminator's perspectives. The generator aims to minimize this loss by generating images that the discriminator mistakes as real, while the discriminator aims to maximize it by accurately distinguishing real from generated images.
λ	A weighting factor that balances the contribution of the L1 loss in the overall objective function of the generator.
$\mathcal{L}_1(G)$	The L1 loss that measures the pixel-wise absolute difference between the generated images and the target images, encouraging the generator to produce accurate reconstructions.
$+\lambda \cdot \mathcal{L}_1(G)$	This term is added only to the generator's loss function to enforce similarity to the target images in the generated output.

2.1.2 CycleGAN

On the other hand, CycleGAN addresses scenarios where paired examples are not available by learning to translate between two unpaired image domains, by continuing the approach of pix2pix. Through the introduction of a Cycle Consistency Loss, CycleGAN ensures that an image can be translated from one domain to the other and back again, retaining its original identity. This allows for effective translation even in the absence of direct correspondences between the source and target domains, making it suitable for a wider range of applications, as in Figure 2.4.

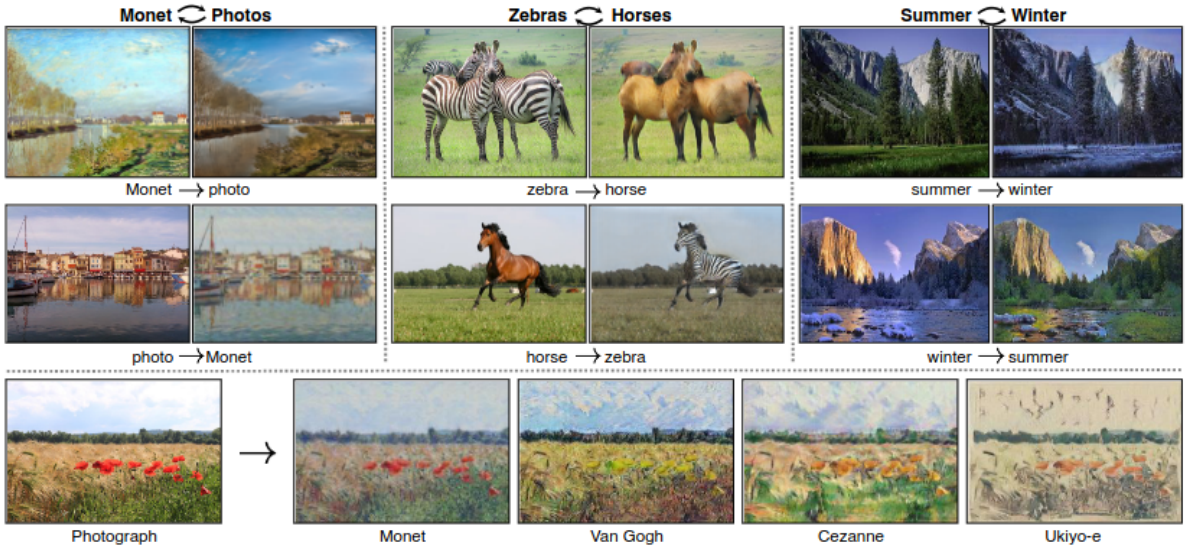


Figure 2.4. Example from the cycleGAN architecture converting unpaired images [5].

CycleGAN consists of two generators and two discriminators, structured into two mirroring GAN setups — each responsible for learning the translation in one direction between the two domains (Domain X to Domain Y, and Domain Y to Domain X), as in Figure 2.5. The generators, labeled

G and F, serve specific roles: G translates images from Domain X to Domain Y, while F performs the reverse, translating images from Domain Y to Domain X. Similar to the approach in pix2pix, each generator's goal is to produce images that are indistinguishable from real images within the target domain. The discriminators, denoted as D_X and D_Y, have their distinct functions as well: D_X discriminates between real images from Domain X and those translated by F, whereas D_Y differentiates real images from Domain Y and those translated by G.

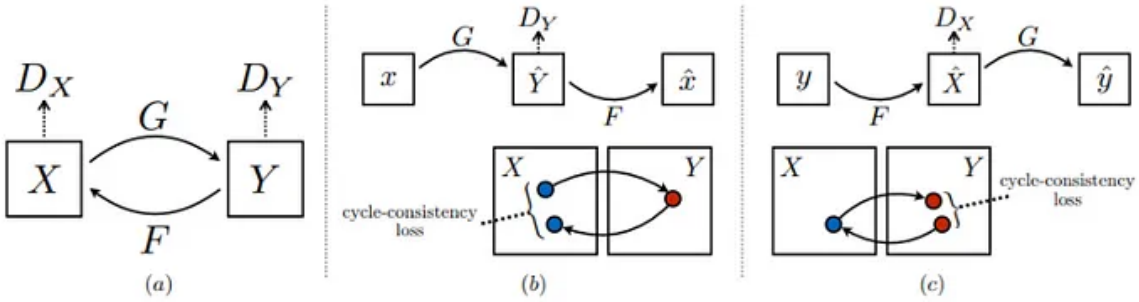


Figure 2.5. (a) The CycleGAN framework comprises two generative adversarial networks, each with a generator and a discriminator. (b) The translation process begins with an image x from domain X, which is passed through generator G to produce the fake image $\hat{y}$, an imitation of domain Y. Discriminator D_Y assesses the authenticity of $\hat{y}$. The translated image $\hat{y}$ is then cycled back using generator F to recreate the original image x, with the fake image denoted as $\hat{x}$. (c) Same process as in (b) is repeated for the image y [5].

An important component for the CycleGAN architecture is the Cycle Consistency Loss. This component ensures that an image translated from one domain to the other can be translated back to the original domain, closely resembling the original image and encourages the model to be able to fully get back to the starting point, as denoted in Equation 2.2.

$$\mathcal{L}_{\text{cyc}}(\mathrm{G},\mathrm{F}) = \mathbb{E}_{\mathrm{x}\sim \mathrm{p}_{\text{data}}(\mathrm{x})}[\|\mathrm{F}(\mathrm{G}(\mathrm{x})) - \mathrm{x}\|_1] + \mathbb{E}_{\mathrm{y}\sim \mathrm{p}_{\text{data}}(\mathrm{y})}[\|\mathrm{G}(\mathrm{F}(\mathrm{y})) - \mathrm{y}\|_1] \tag{2.2}$$

$\mathcal{L}_{\text{cyc}}(G, F)$	The cycle consistency loss that ensures the mappings G and F are consistent with each other by enforcing that an image translated to the target domain and then back to the original domain should look like the original image.
$\mathbb{E}_{x\sim p_{\text{data}}(x)}[\|\|F(G(x)) - x\|\|_1]$	The forward cycle consistency loss which measures the average absolute difference (L1 norm) between the original image x and the image obtained after passing x through the generator G and then the inverse generator F.
$\mathbb{E}_{y\sim p_{\text{data}}(y)}[\|\|G(F(y)) - y\|\|_1]$	The backward cycle consistency loss which measures the average absolute difference (L1 norm) between the original image y and the image obtained after passing y through the generator F and then the inverse generator G.
G	The generator that maps images from domain X to domain Y.
F	The generator that maps images from domain Y to domain X.
$\|\| \cdot \|\|_1$	The L1 norm used to measure the pixel-wise absolute differences between images, promoting accurate reconstructions.
$x \sim p_{\text{data}}(x)$	The distribution of images in the source domain X.
$y \sim p_{\text{data}}(y)$	The distribution of images in the target domain Y.

This cycle consistency is fundamental to CycleGAN's ability to learn meaningful translations without paired data, as it ensures a balanced translation and prevent the learned mappings from contradicting eachother. It effectively means that $G(F(x)) \approx x \wedge F(G(y)) \approx y$.

This results in a final objective similair to the one of pix2pix, but with an extended generator and discriminator part, as in Equation 2.3.

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_y} \mathcal{L}(G, F, D_x, D_y) \tag{2.3}$$

Where

$$\mathcal{L}(G, F, D_x, D_y) = \mathcal{L}_{\text{GAN}}(G, D_y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_x, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) \tag{2.4}$$

Both CycleGAN and pix2pix architectures represent significant advancements in conditional GAN research, offering paths for generating realistic images across different domains. Whether dealing with paired or unpaired data, these frameworks provide powerful tools for researchers and practitioners looking to bridge the gap between different image modalities, such as RGB and thermal imaging, providing the path for further experimentation.

2.2 Dataset

To proceed with further experimentation, a domain-paired dataset is required. The dataset used for the experiment includes recordings of 17 subjects seated in a simulated driver's seat, undergoing a complete simulation drive. The dataset captures RGB images at approximately 30 frames per second (fps), while the thermal images are recorded using a thermal camera operating at less than 9 fps. This makes it a pertinent case for this study. The thermal images

cover a temperature range from -20°C to 300°C, and are scaled to the 0-255 range. The data collection setup included four RGB cameras and one thermal camera. The arrangement of these components is illustrated in Figure 2.6. The cameras each save individual frames with a timestamp. All cameras are connected to the same desktop, providing the timestamps for the images.

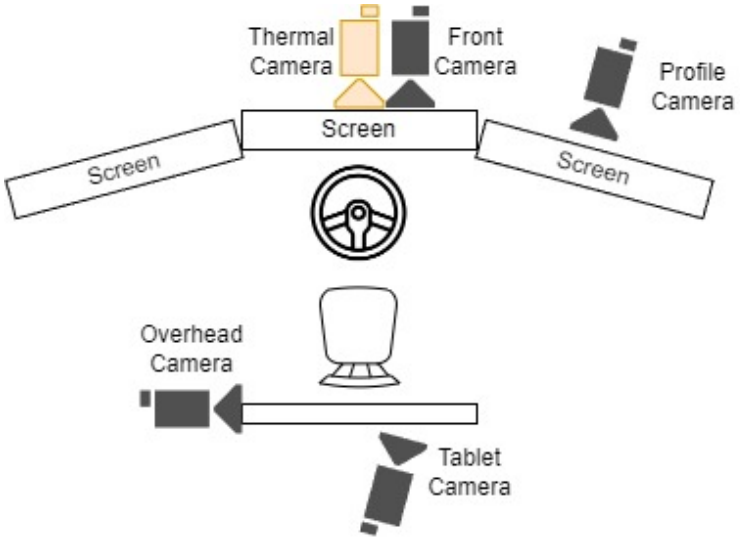


Figure 2.6. The Driving Simulator is equipped with a driver's seat, an interactive steering column, and three screens that display the simulated driving environment. It features four cameras, a microphone, and a thermal camera for sensing purposes.

The dataset, originally designed for use in driver feature extraction projects, was not initially intended for this experiment. However, it has proven to be well-suited for this study.

During the dataset creation, several factors affected its consistency. Cameras were unintentionally moved between subjects, some subject frames included test personnel, and contamination is part of the dataset, such as different furniture appearing in the background for some subjects actively making some classification tasks easier.

RGB Cameras 1-4: Logitech Brio Camera [7]
Four RGB cameras are positioned around the test subject, as illustrated in 2.6. Each camera operates at a resolution of 1920x1080p, with a 90-degree field of view and a frame rate of 30 fps.

Sensor Resolution	Field of View	Frame Rate
1920x1080p	90 deg	30 fps

Thermal Camera: Seek Thermal Micro Core Starter Kit [8]
The dataset consists of an additional camera with thermal capabilities, mounted parallel to Camera 1 as shown in Figure 2.6. The thermal camera has a resolution of 200x150p, 61 deg field of view, and a frame rate of < 9Hz.

Sensor Resolution	Field of View	Frame Rate
200x150p	61 deg	< 9Hz

2.2.1 Camera viewpoints

Thermal Camera is mounted parallel to Camera 1 to capture the thermal signatures from the subject's face.

Camera 1 (front) is mounted directly above the center screen and captures the subject's face. This camera is meant to capture gaze features and a full frontal view of the eyes and face.

Camera 2 (overhead) is mounted directly above the subject's head and captures body and hand positions.

Camera 3 (profile) is also mounted behind the steering wheel, but in addition to the driver's face, provides another view of the subject's body posture. In particular, this view is meant to capture auxiliary information on the driver's entry into the seat and general posture changes and movement, for possible analysis of driver behavior when entering the vehicle.

Camera 4 (tablet) is mounted behind the subject's right shoulder to capture potential hand attention to a given task on a tablet.

Examples of each camera view are shown in Figure 2.7. Only the presented subject denoted as *0* has given consent to be utilized for display in this report.

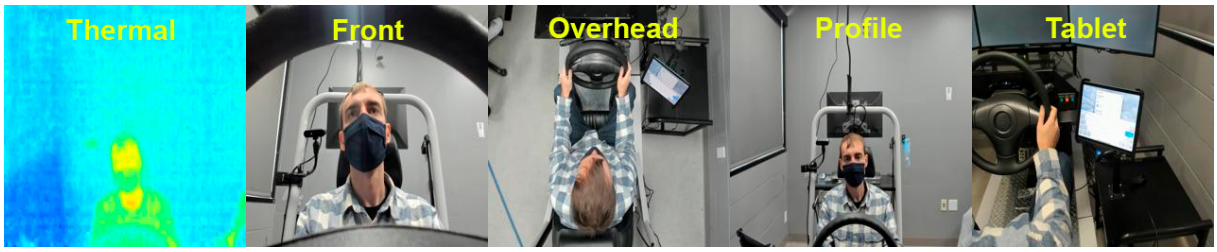


Figure 2.7. The five views of the driver: overhead view, face view, thermal view, profile view, and tablet view. The first three views represent the driver-center views, while the last two views represent a view for entry pose analysis and tablet-related features.

2.2.2 Multi-sensor synchronization

In order to establish a reliable ground truth for the generation of frames, synchronizing the views captured by different cameras is important. Each camera provides a unique perspective of the scene, and aligning their timestamps ensures that the captured events are accurately correlated across all viewpoints. As the timestaps are generated by a desktop possibly affected by latency and load of frames, all timestaps may not be fully accurate. However, when investigating the data, it aligns for the purpose of this experiment.
The chosen synchronization methodology employs a binary search algorithm to align timestamps of images captured by different cameras, included in the process described in Figure 2.8.
Binary search operates on the principle of dividing the search space in half repeatedly until the desired timestamp is located, as portrayed in Figure 2.9.

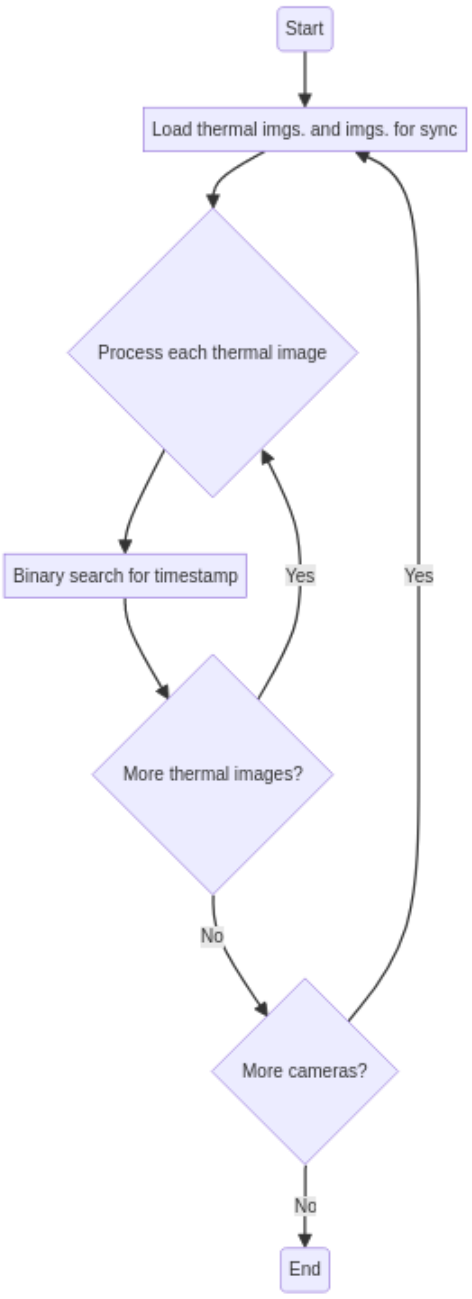


Figure 2.8. Synchronization procedure utilizing Binary Search.

Algorithm:
Start in the Middle: The search begins by examining the image in the middle of the chronologically ordered collection. Since the images are sorted by timestamp, this middle image represents the midpoint of the entire time range covered by the images.
Comparison: The timestamp of the middle image is compared to the target timestamp (the timestamp of the image you're looking for).
Narrowing Down: Based on this comparison, it's determined whether the target image would be located to the left or right of the current image in chronological order. This halves the search space.
Repeat: The search process is then repeated with the half of the collection where the target image is likely to be found. This process continues recursively until the target image is located.
By repeatedly halving the search space, binary search quickly narrows down the possible locations of the target image, making it a highly efficient method for image retrieval based on timestamps. Once the algorithm can no longer divide the search space further, it selects the timestamp that is closest to the target timestamp within the remaining range [9].

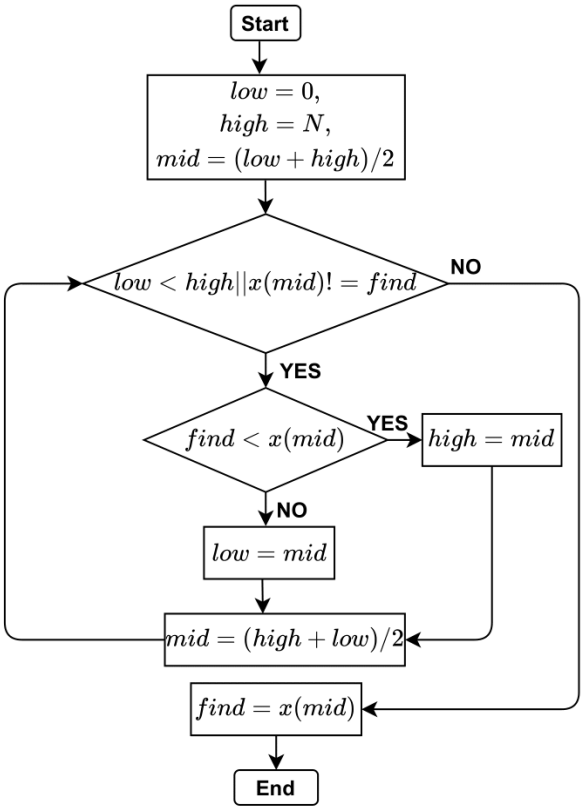


Figure 2.9. Flowchart for Binery Searh algorithm [10].

2.3 Experimental evaluation

This chapter will test the discussed methods by using different input layouts of the dataset, as in Figure 2.10. The metric used to evaluate the generated quality will be L1 loss, as this is the metric used by the corresponding papers and there will be a ground truths for all tests. All tests performed with standard settings recommended from the pix2pix [4] and CycleGAN [5] papers, unless otherwise stated. Implementations with inspiration from [11, 12, 13, 14].

(a) Thermal ground truth. (b) Front-View. (c) Four-View, Tessellated. (d) Four-View, Stacked.

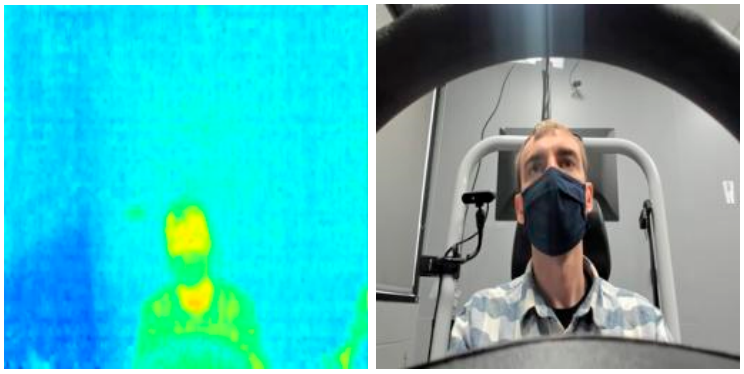
Figure 2.10. Different inputs for the experiments. All were evaluated on their potential for generating an image similar to the thermal ground truth.

As the dataset comprises individual subjects it's chosen to keep initial testing on an individual basis, as different subjects can have different RGB and thermal characteristics. Therefore, a

model is trained for each subject. From each subject a collection of 500 thermal + RGB image synchronous groups is created, to align with the subject that holds the minimum amount of data. In each experiment, an allocation of 80 % for training, 10 % for validation, and 10 % for testing has been used. All training curves are displayed up to 20,000 steps (≈ 40 epochs), as there are no significant changes observed beyond this point, unless specified otherwise. All examples provided pertain to the subject identified as 0, who has granted permission for the use of their photographs.

2.3.1 Front-View

As the wanted output is a thermal image of the front view, naturally the Front-View RGB is selected as the first testing point. Thermal Front-View as the ground truth and the Front-View RGB image is selected as the input to generate the thermal images. It's anticipated that an RGB to thermal image mapping can be created from a Front-View RGB image, as this is the most corresponding image. An example of an element in the dataset is seen in Figure 2.11. Experiments will be conducted for both pix2pix, CycleGAN and a combination of all subject data.



(a) Thermal ground truth. (b) Front-View.

Figure 2.11. Input style for Front-View experiment.

pix2pix

First experiment is conducted using pix2pix, achieving training metrics in Figure 2.12. These metrics illustrate the model's performance throughout the training process, thereby establishing whether the model potentially fits the intended criteria or not.

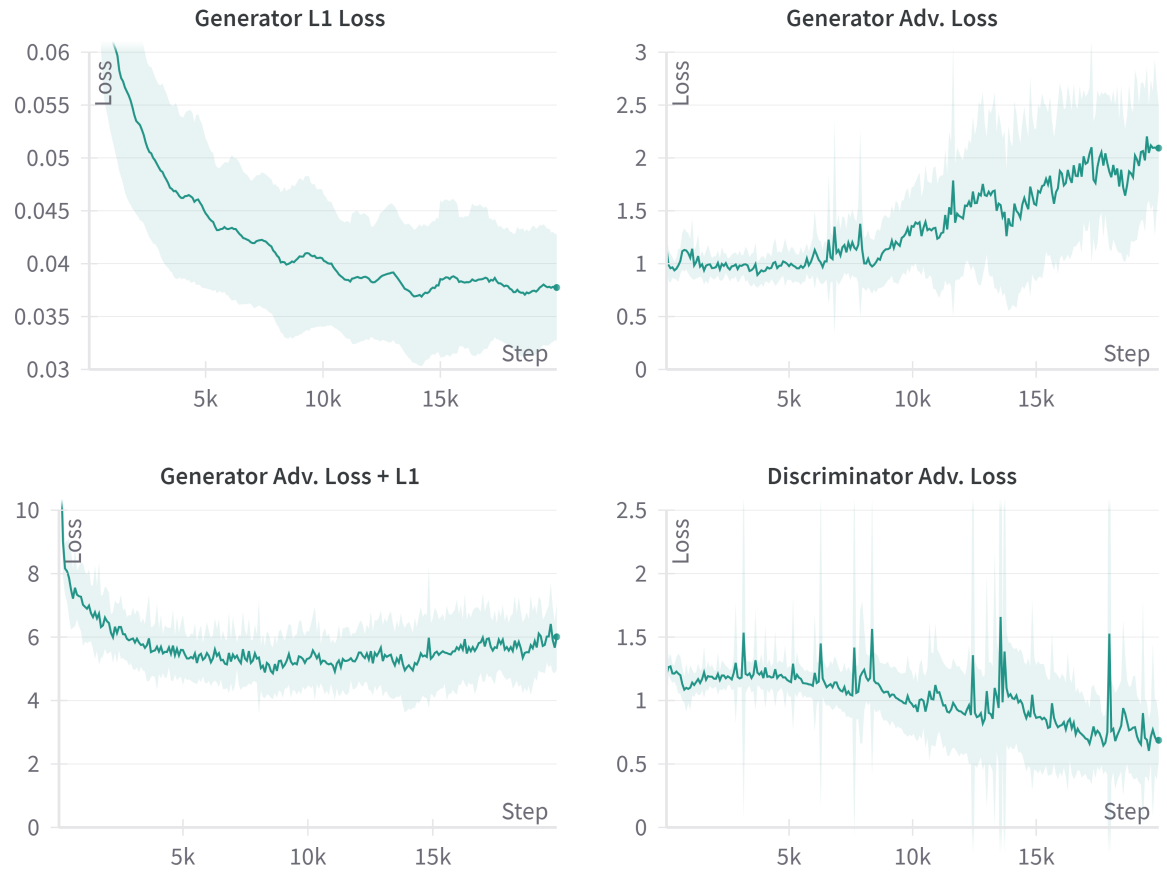


Figure 2.12. Generator L1, adversarial, adversarial + $\lambda \cdot$ L1 loss and discriminator adverserial loss for the front single subject test. Training curves showing average values and standard deviation over the 17 subjects.

When looking at this data, it's observed that the L1 loss of the generator is steadily going down and improving, resulting in a minimum average loss of 0.0363 (σ .0047), indicating that the generator is producing images close to the ground truth. However, when looking at the individual adversarial loss of the generator and discriminator, there is an indication of a winner to the minimax game. The discriminator is increasingly improving while the generator is declining, indicating that it's too easy to differentiate between real and generated images. The validation accuracy demonstrates signs of overfitting between steps 15,000 and 20,000, corresponding to when the generator begins to visibly fall behind the descriminator. The model chosen for testing is based on the lowest validation loss achieved, which for the example of the average validation loss in this instance is 0.697 (σ 0.012) at step 14,000, as shown in Figure 2.13.

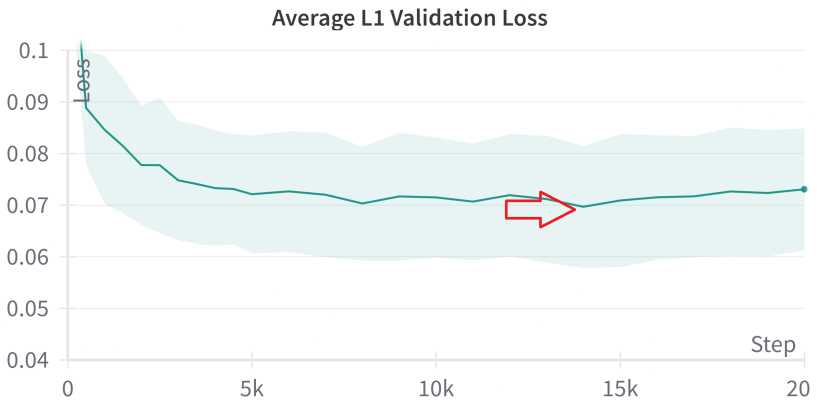


Figure 2.13. Validation L1 loss for the Front-View test, resulting in a lowest score of average score of 0.697 (σ .012).

However, despite the relatively high losses, when looking at the images, the similarity is fairly close as in Figure 2.14. Figure A.1 in Appendix A displays different iterations starting from generated noise to an approximation, showcasing the iterative process of the minimax game.

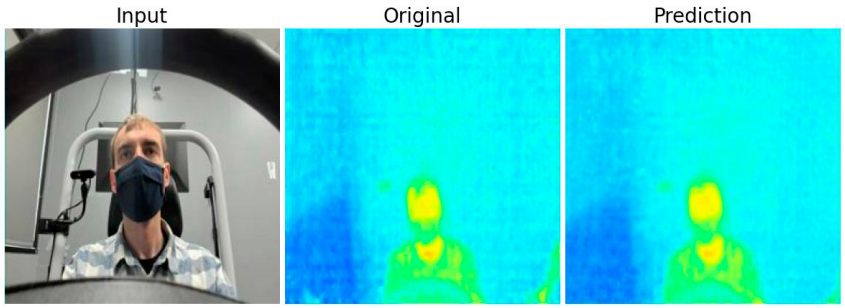


Figure 2.14. Single-Subject Front-View Prediction Example. L1 Loss of 0.0625.

The final testing results, as depicted in Table 2.1, reveal notable variations in the performance metrics among the subjects tested. These disparities can be attributed to several factors, which may include but are not limited to, different subjects possessing distinct thermal characteristics and varying camera setups that can influence the outcome of the tests.

Subject	Test/Average L1 Loss
0	0.048
1	0.072
2	0.079
3	0.070
4	0.076
5	0.049
6	0.088
7	0.067
8	0.066
9	0.071
10	0.065
11	0.064
12	0.062
13	0.084
14	0.069
15	0.065
16	0.054
Average:	**0.0676 (σ 0.0106)**

Table 2.1. Test/Average L1 Loss by subject for pix2pix front camera only. Worth noting for subject 0 is that this is the only subject wearing a mask. Full subject list to indicate the loss range.

Different subjects may have unique thermal profiles due to variations in factors such as clothing, movement of the camera, and ambient conditions during the tests. For example, subject 0, who was wearing a mask, showed a distinctively lower L1 loss compared to others, suggesting that masks might make the thermal prediction easier, as it removes details and thus the performance of the model. The relocation of the camera and variations in subject height present known challenges for maintaining a consistent field of view. This often results in compromised data quality, as illustrated in Figure 2.15, where obstructions such as the steering wheel obscure significant portions of the subject's face in the thermal view, but not in the RGB view.

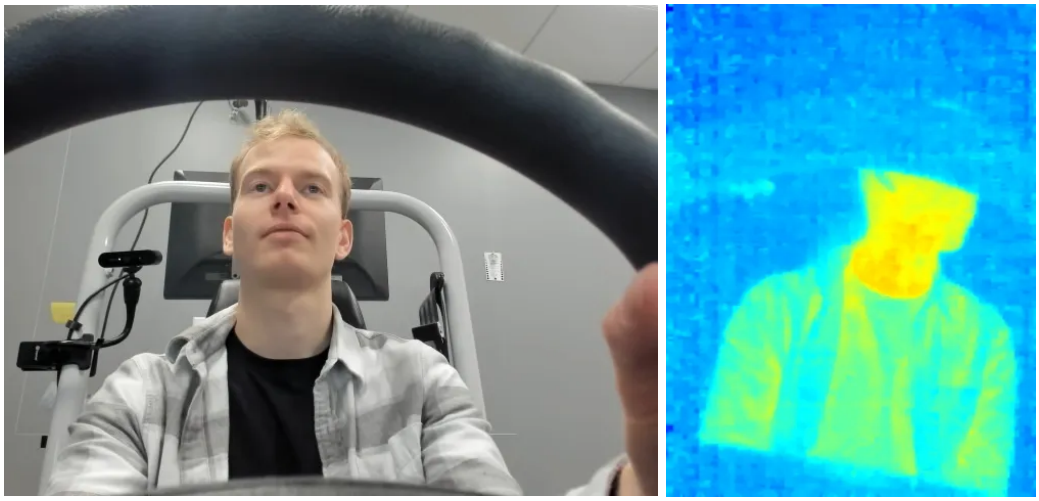


Figure 2.15. Obstacles can occur in the dataset with a frequent one being the steering wheel covering large parts of the face in the thermal point of view, potentially confusing models. This also illustrates the possible mismatch when sensors operate at different rates, as it is possible that the temporally-nearest measurement to a given instance may have taken place before a significant action for one sensor, and after the action for another. In the above example, the driver has abruptly moved his face to front camera looking position, captured by the RGB camera; however, the thermal camera has not yet processed another signal to capture this motion, and is still capturing the subject looking to the side. This given subject has not been utilized for this experiment.

CycleGAN

The same experiment has been conducted using CycleGAN. Results are shown for 100,000 steps ($\approx$ 200 epochs). The expectation of using CycleGAN is that it can perform a better generalization, as it's domain-focused and not paired translations as pix2pix.

Analysis of the training characteristics reveals that the Cycle Consistency Loss for the RGB domain (generator F) underperforms relative to the thermal domain (generator G) as displayed in Figure 2.16. This discrepancy is likely due to the higher detail complexity in RGB images, which makes the RGB domain more challenging to predict accurately compared to the less detail-intensive thermal images.

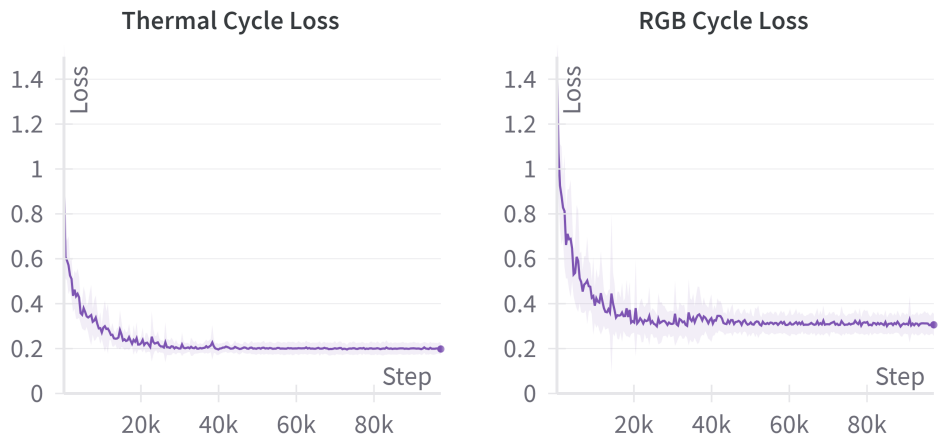


Figure 2.16. Cycle Consistency Loss for the thermal and the RGB training inputs. Training curves showing average values and standard deviation over the 17 subjects.

Further inspection of the CycleGAN training dynamics, Figure 2.17, Generator G, tasked with synthesizing thermal images, demonstrates a steady decline in adversarial loss, reflecting its growing proficiency in creating images that are increasingly difficult for its paired Discriminator X to classify as fake. Meanwhile, Generator F, which generates RGB images, shows higher variability and overall loss, indicating a struggle to produce realistic RGB images convincingly.

Conversely, the corresponding discriminator for the RGB domain, Discriminator Y, exhibits a higher adversarial loss than Discriminator X, suggesting that it has more difficulty distinguishing between real and generated RGB images. This could imply that even though Generator F is less effective at producing realistic images compared to Generator G, Discriminator Y is also less effective at detecting the forgeries, making the task seem relatively more complex for the RGB domain.

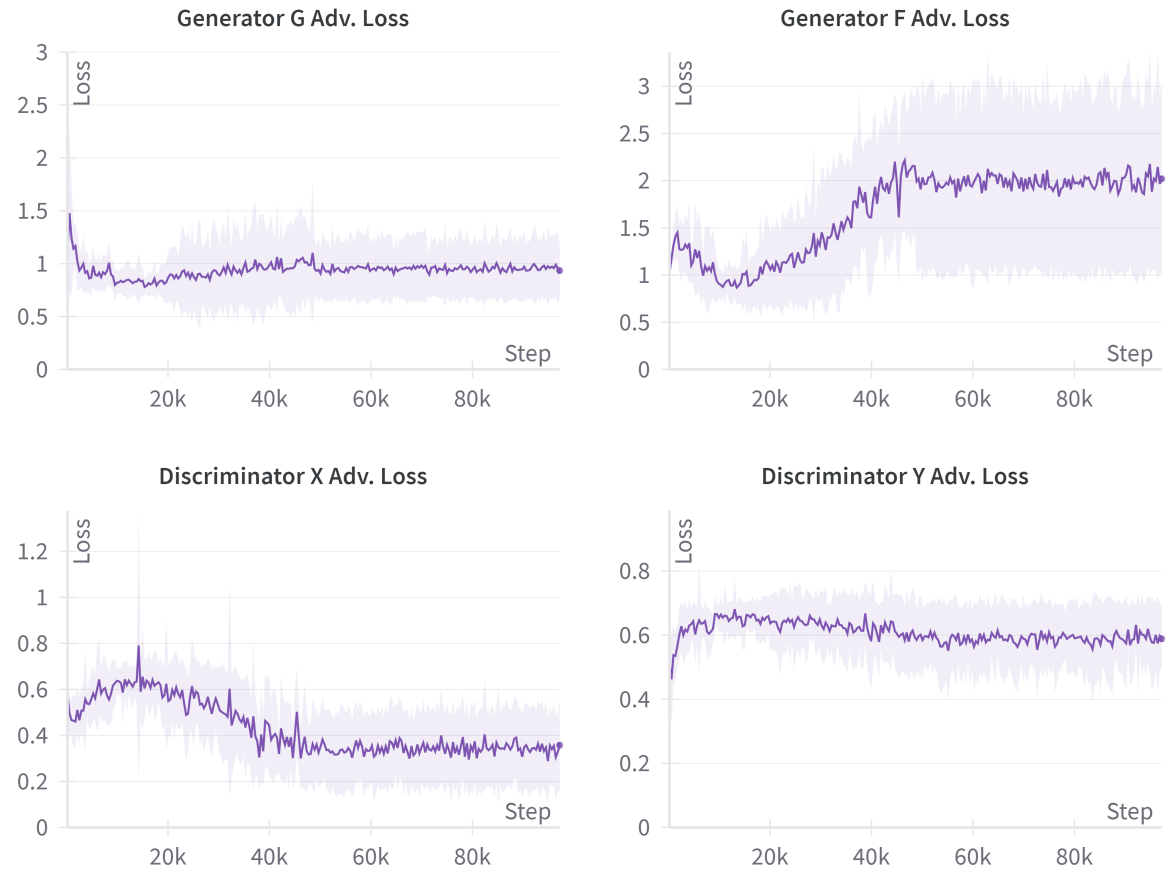


Figure 2.17. Adversarial losses for generators G and F, and discriminators X and Y, over 100 k steps, illustrating the competitive training dynamics in CycleGAN. Training curves showing average values and standard deviation over the 17 subjects.

The L1 loss data for individual samples are summarized in Table 2.2. The variance in performance across subjects, as indicated by the standard deviation, provides further insights into the model's consistency and underscores the challenge of generalizing across varied inputs. Notably, the average L1 loss computed across all subjects offers an overall measure of the model's predictive accuracy.

Subject	Test/Average L1 Loss
0	0.165
1	0.238
2	0.128
3	0.213
4	0.255
5	0.070
6	0.097
7	0.168
8	0.110
9	0.151
10	0.104
11	0.083
12	0.194
13	0.247
14	0.205
15	0.143
16	0.226
Average:	**0.1644 (σ 0.0585)**

Table 2.2. Test/Average L1 Loss by subject for the CycleGAN front camera only. Full subject list to indicate the loss range.

In comparison to the pix2pix framework, the CycleGAN's performance exhibits less precision, with an average L1 loss of 0.1644, as opposed to the more favorable average loss of 0.0675 observed in pix2pix. This disparity in accuracy can be attributed to the inherent challenges in cycling a precise RGB image—a process intrinsic to CycleGAN but absent in the pix2pix architecture, which likely accounts for the latter's enhanced performance. An illustrative CycleGAN prediction is presented in Figure 2.18, demonstrating a well-made mirrored prediction. While this example stands out as one of the more successful outcomes within the CycleGAN tests, it does not achieve the same level of fidelity seen in the pix2pix predictions. Given these findings, the CycleGAN architecture will not be further explored in this work.

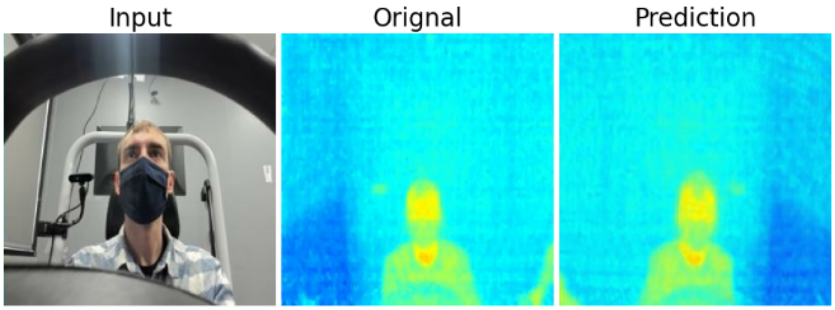


Figure 2.18. Single-Subject Front-View CycleGAN prediction example. L1 loss of 0.163.

Training on multiple subjects, combined dataset

In pursuit of deploying the model in practical applications, a step of the way to achieve this goal is to ensure it generalizes effectively across different subjects. To this end, it's wanted to assess

its performance using a consolidated dataset that encompasses samples from all subjects. Due to limitations of used training infrastructure, a representative subset of 5,000 randomly selected samples from the collective 8,500 available is selected.

The same pix2pix training characteristics as in the previous Front-View experiment are present, but with a clear tendency of declining performance. Figure 2.19 displayes a greater generator L1 loss while training, as well as the discriminator having a lower adversarial loss, indicating that the generator is performing worse.

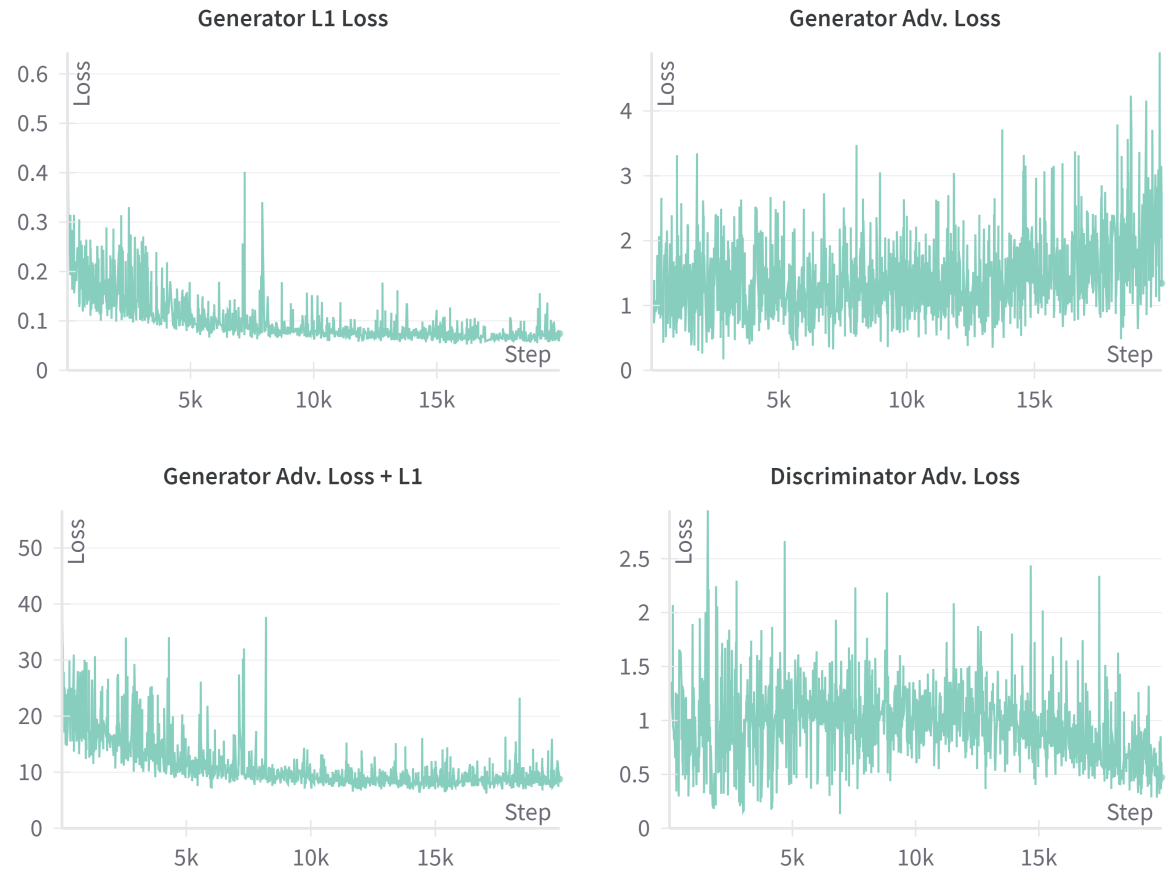
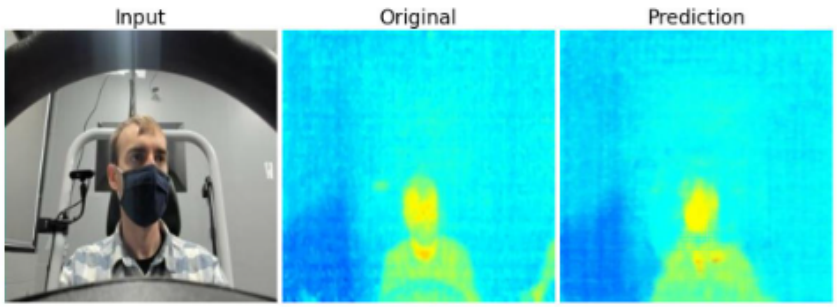
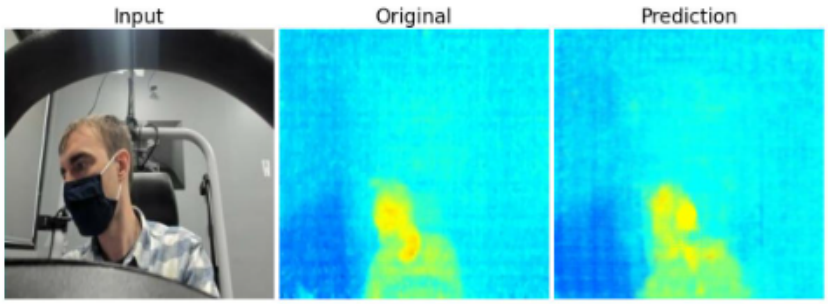


Figure 2.19. Generator L1, adversarial, adversarial + $\lambda \cdot$ L1 loss and discriminator adverserial loss for a combination of all subjects.

Figure 2.20a & b reveal that the model struggles to generate accurate thermal image predictions, likely due to the increased diversity within the dataset.



(a)



(b)

Figure 2.20. Multi-Subject Front-View Prediction Example. L1 loss of 0.1002 (a) and 0.162 (b).

The example from Figure 2.20b has an L1 loss of .1002. This however, has a large potential of being the result of the given example subject wearing a face mask, as the 16 other subject do not wear a face mask, and the model struggles to generalize with this. When looking at the predicted data, this appears to be the case as most outliers belong to subject 0, with the face mask, while other subjects still not being as well predicted as training and testing on a single subject. It's important to note that each subject has unique seat settings and height, and the cameras were slightly moved, albeit unintentionally, between some subjects. This variability introduced additional complexity for the model, potentially affecting its performance in the combined subject space. As there is no permission to show the the data of the other subjects, this can not be further discussed in this report.

Using the same training procedures and architectures as before, the best validation accuracy of .106 is achieved, resulting in a further test accuracy of .112. A similar average performance of .103 is acchieved testing without subject 0.

2.3.2 Four-View

As the data is available from different angles covering the same subject, investigation as to how the thermal generation would benefit from using more camera views in a single image, potentially providing a better generalization, is started. The two views as in Figure 2.10c, Tessellated, and 2.10d, Stacked, will be investigated. As the performance of CycleGAN and the combination of all subjects to one dataset did not perform as well as the pix2pix architecture and single subject testing, these are excluded from further testing in this section. The pix2pix training characteristics for both the Tessellated and Stacked can be seen in Figure 2.21, along with a comparison to the Front-View only training characteristics.

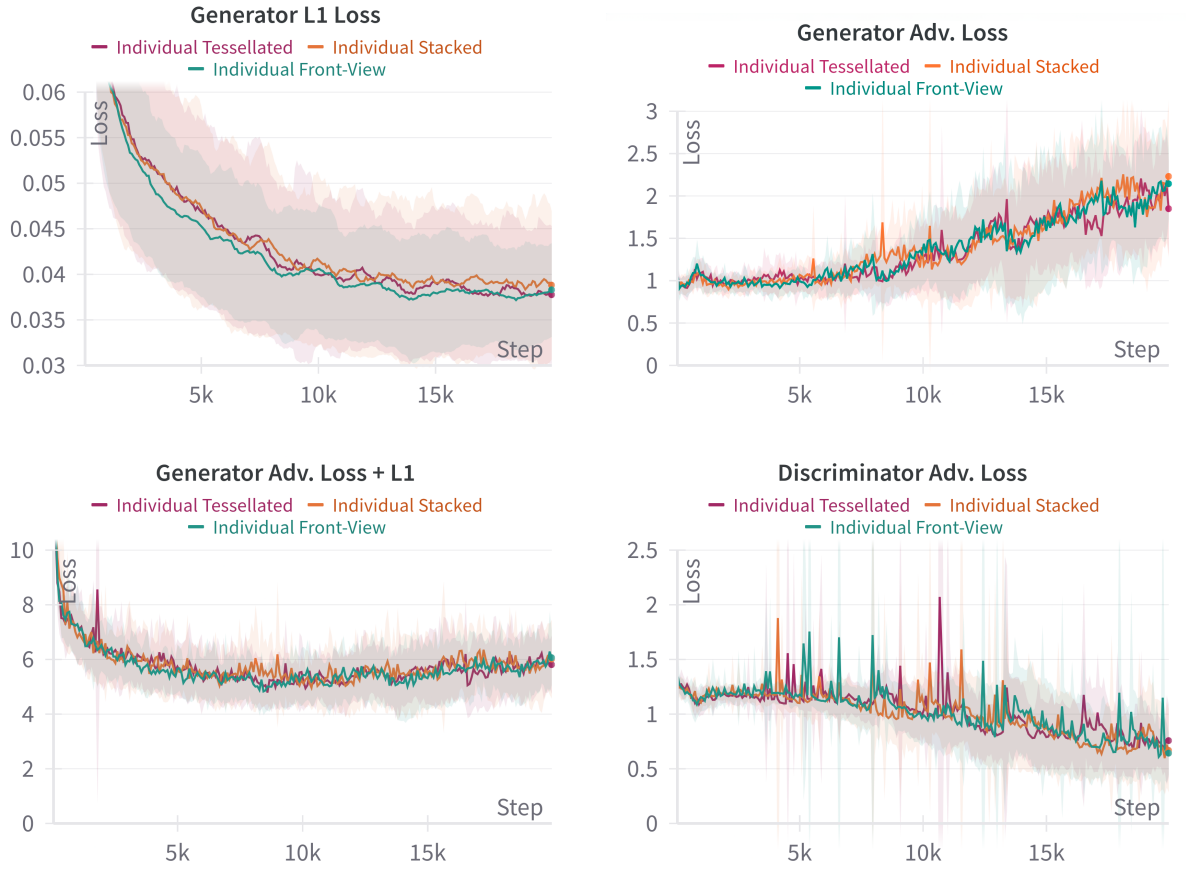


Figure 2.21. Generator L1, adversarial, adversarial + $\lambda \cdot$ L1 loss and discriminator adversarial loss comparison for the Four-View Tessellated, Stacked and the Front-View. For better comparison, running average [15] has been applied with a smoothing factor of 10 to the Generator L1 Loss graph. Training curves showing average values and standard deviation over the 17 subjects.

It's observed that these perform very similarly both in training performance and tendencies, indicating that the extra detail did not change anything significantly for the training process. The discriminator is still winning the minimax game, and generator seems to be declining. However, the generators L1 loss for the Four-View seems to be slightly higher, indicating that it's not as easy to fit to the training data. This has shown to be beneficial when investigating the validation and test loss in Figure 2.22, indicating better generalizability for the thermal prediction.

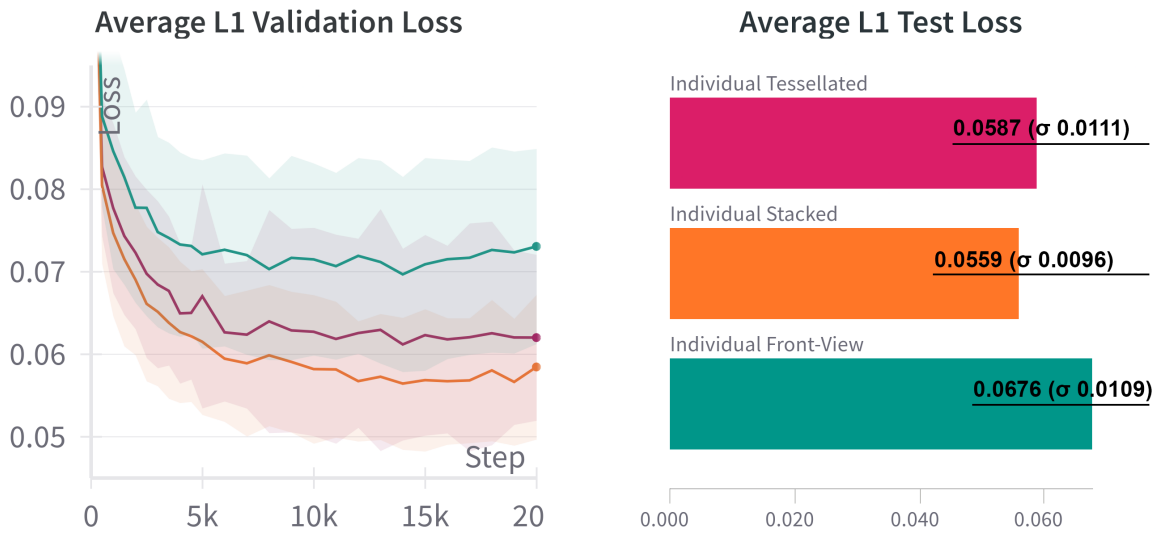


Figure 2.22. Compared validation loss and test loss for Four-View Tessellated, Stacked and the Front-View.

Prediction examples in Figure 2.23, indicating a close similarity between the original and prediction for both, but with a clear indication of the Stacked (b) as being the most accurate.

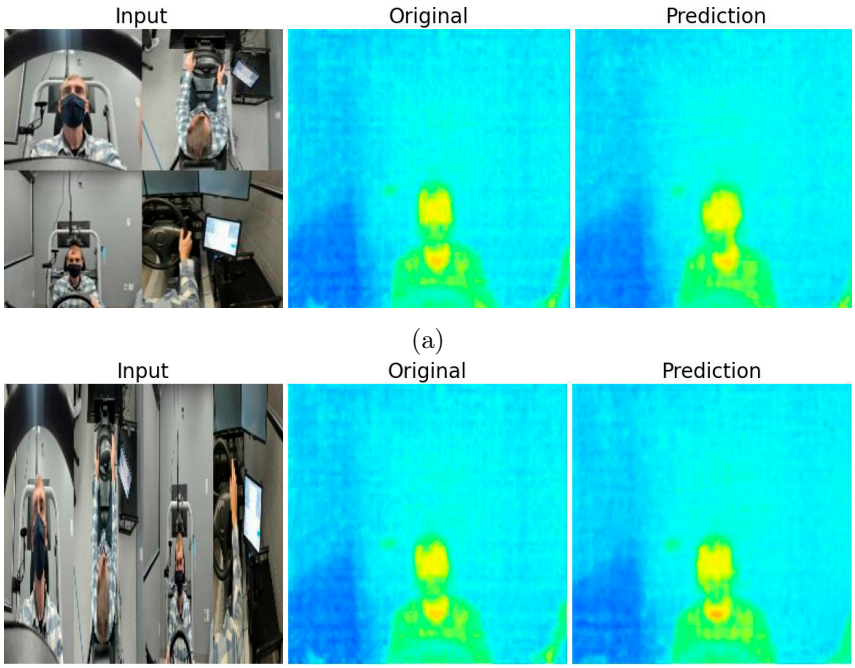


Figure 2.23. Four-View, (a) Tessellated Prediction Example, (b) Stacked Prediction Example.

2.3.3 Summarized results

Method	Average Test L1 Error	Standard Deviation
CycleGAN	0.1644	0.0633
pix2pix	**0.0676**	**0.0585**

Table 2.3. Comparison of Model Architectures when training Front-View.

Dataset	Average Test L1 Error	Standard Deviation
Front-View	0.0676	0.0106
Four-View, Tessellated	0.0587	0.0109
Four-View, Stacked	**0.0559**	**0.0093**

Table 2.4. Comparison of input style; average performance across 17 subjects.

Dataset	Average Test L1 Error	Std. Deviation
Single-Subject Training	**0.0676**	**0.0106**
Multi-Subject Training	0.1116	0.0186

Table 2.5. Comparison of single vs. multi-subject training on Front View. Single-subject is the average of 17 subjects while multi-subject is the average of 17 identical runs.

2.4 Concluding remarks

In exploring the generation of thermal frames from RGB frames using cGANs, this work utilized pix2pix and CycleGAN as the primary frameworks. The Stacked generation approach proved most successful, emphasizing the importance of spatial relationships. Despite these advancements, the model's generalization between subjects remains the worst performer, warranting continued efforts for improved adaptability across diverse scenarios so that singular models may be trained and deployed, and motivating research into the potential of small-data fine-tuning for model customization to individual drivers. While these models, especially pix2pix, have demonstrated well performing, the integration of other sophisticated methods could potentially enhance the model's accuracy and robustness.

Worth noting for the pix2pix runs is that they are of a minimal training time, with the best performing run *Four-View, Stacked* only averaging 8 min and 2 seconds (σ of 7 s), as displayed in Table 2.6, making it relevant in a use-case scenario.

Dataset	**Average Training Time**	**Std.**
Front-View	7 min. 55 s.	10 s.
Four-View, Tessellated	7 min. 54 s.	5 s.
Four-View, Stacked	8 min. 2 s.	7 s.
Combined	23 min. 55 s.	4 s.
CycleGAN	31 min. 31 s.	37 s.

Table 2.6. Comparison of Input Style; average training time across 17 subjects. All of the above was trained on a NVIDIA GeForce RTX 3090. The combined was re-trained 17 times for more accurate comparison.

Use of inbetweening with conditional GANs

The concept of inbetweening, or frame interpolation, combined with a conditional GAN framework poses an interesting potential advancement. Such integration could facilitate the generation of intermediate frames not only with enhanced visual quality but also with better temporal coherence between successive frames. Although this approach was not implemented in the current study, it could potentially address the challenges related to the generalizability of the pix2pix model and to generating large sequences of frames as noted in Google's inbetweening paper [16]. By exploring scalable solutions that combine the strengths of inbetweening and conditional GANs, longer video sequences could potentially be generated while maintaining temporal coherence and visual integrity, as a model can struggle to predict non-logical movements between frames [16, 17]. As an example, a given 25-year-old male human can reach an object $31\text{cm} \pm 3\text{cm}$ away in 226ms [18], advocating that sudden non predictable movements can happen inbetween frames, benefitting from the use of a corresponding source.

Evaluation Metrics

During this work, it's chosen not to apply more common image generative metrics such as the Inception Score, commonly used in scenarios where no ground truth data is available, because the dataset included labelled pais of thermal and RGB images [19]. Instead, the use of the L1 loss metric was deemed sufficient as it aligns with the evaluation methods used in the pix2pix framework and the initial results were promising. While this study was primarily explorative in nature, focusing on the analysis of individual frames rather than complete video sequences, further insights into the temporal consistency and visual quality of the generated frames could be gained by exploring additional metrics. One such metric is the Frechet Video Distance (FVD), which offers a comprehensive assessment of model performance across video sequences [20]. Although the generation of full video sequences was not deemed necessary for this phase of the research, employing FVD in future evaluations could provide deeper insights into how well the generated frames align over time, thus offering a more complete understanding of the model's capabilities in handling dynamic content.

Camera Placement and Data Quality

The strategic placement of cameras is essential for effectively pairing thermal and RGB images, as illustrated in the example shown in Figure 2.24. This example highlights how the alignment and positioning of cameras significantly impact the quality and utility of the captured data.

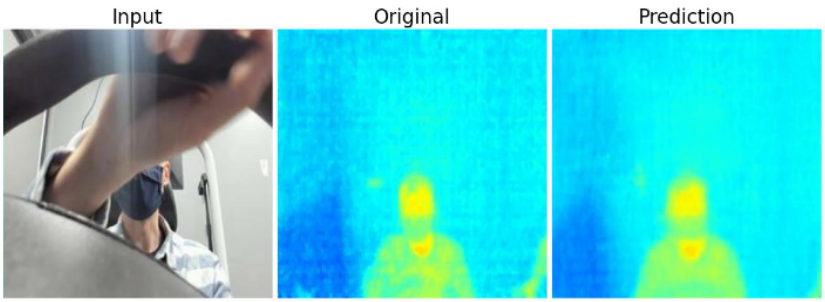


Figure 2.24. Thermal prediction example from the front camera, highlighting imperfections in camera placement. The image shows how one camera captures the arm in its field of view while the other does not, which could lead to potential confusion for a given model prediction.

The figure underscores a prevalent challenge where improper camera positioning can result in

'areas of confusion' in the captured images. In this specific case, two cameras were employed to record RGB and thermal data from the same position; however, their differing viewing angles caused a significant discrepancy in coverage. Notably, the arm of the subject is clearly visible in the field of view of the RGB camera but is partially or entirely missed by the thermal camera. This inconsistency complicates the interpretation of data, potentially affecting the accuracy of thermal predictions, as it can confuse the model.

Similarly, the thermal camera has been moved in between subjects, causing the thermal imaging to differentiate more than necessary, making a generalization between subjects unnecessarily more complicated.

Model Enhancements
The Stacked Four-View architecture has shown better performance during validation and testing phases, yet it exhibits higher training losses, suggesting potential overfitting tendency as others performed better while training but worse during inference. Employing a deeper generator could help mitigate this issue by improving the model's ability to generalize better to unseen data. Further, replacing the fused input image with a dedicated fusion network might not only preserve and process RGB data streams more effectively but also enhance the precision of the synthesized thermal images, despite the added complexity this might introduce. Expanding the generator's capacity and employing a larger, more complex model could further improve the detailed capture of thermal imagery nuances. Additionally, it could be interesting to explore the use of diffusion models, which have demonstrated significant improvements in generating high-quality, realistic images and could potentially provide more accurate and detailed thermal images [21].

In summary, the generative approach shows potential to address the missing frames problem (caused by sensor frame rate mismatches and intermittent failures), providing a means for higher frequency driver state monitoring for enhanced intelligent vehicle awareness and rapid, safe decision-making.

Driver Activity Classification Using Generalizable Representations from Vision-Language Models

3

This work has been accepted at the Computer Vision and Pattern Recognizion (CVPR) Vision and Language for Autonomous Driving and Robotics Workshop scheduled for June 18 2024. The work has been carried out in collaboration with PhD candidate Ross Greer, a member of LISA.

An article of the name *Driver Activity Classification Using Generalizable Representations from Vision-Language Models* can be found in Appendix D. It is advisable to read the article before engaging with this chapter, as this chapter build upon the article's understanding.

This chapter aims to explore the potential of using Vision-Language models for predicting in-car cabin activities—such as drinking, talking on the phone, and texting—utilizing multiple camera views without the need for fine-tuning already existing generalizable models while applying minimal post-processing. It highlights the flexibility of Vision-Language models, demonstrating their applicability to a broad range of tasks beyond mere linguistic functions.

AI City Challenge: Naturalistic Driving Action Recognition

This chapter applies theoretical concepts to a practical scenario presented by the 2024 AI City Challenge at CVPR, specifically Track 3: Naturalistic Driving Action Recognition. This track addresses the critical issue of distracted driving, which is a significant safety concern in the United States, responsible for approximately eight fatalities daily. [22]

The challenge utilizes synthetic naturalistic video data that captures driver actions from multiple camera angles within the vehicle, aiming to identify and mitigate distracted behaviors such as using a phone or eating. It offers a dataset depicting drivers engaged in 16 different tasks that could distract them from driving, as in Table 3.1, captured in a real-world-like environment, from the angles pictured in Figure 3.1. All data was recorded inside a stationary car, with participants performing a series of staged actions. The dataset comprises two parts, a labeled part A and an unlabelled part B, with part A being utilized in this work. It includes approximately 62 hours of footage featuring 69 subjects, recorded at 30 frames per second. Part A comes in 7 folders each containing a set amount of subjects. This split will be utilized during experimentation.

Figure 3.1. Illustration of multi-perspective in-cabin camera views for monitoring driver behavior under the class '0: Normal Forward Driving'. (1) Dashboard view. (2) Rear-view. (3) Side view.

Class	Activity Label	Dist. %	Number of Used Frames
0	Normal Forward Driving	59.01	1176316
1	Drinking	1.49	29781
2	Phone Call (right)	2.78	55732
3	Phone Call (left)	2.97	59231
4	Eating	3.29	65601
5	Text (Right)	3.44	68655
6	Text (Left)	3.56	70878
7	Reaching behind	1.40	27983
8	Adjust control panel	2.42	48320
9	Pick up from floor (Driver)	1.31	26121
10	Pick up from floor (Passenger)	2.15	42825
11	Talk to passenger at the right	3.52	70215
12	Talk to passenger at backseat	3.46	68986
13	Yawning	1.87	37217
14	Hand on head	3.45	68871
15	Singing or dancing with music	3.85	76665

Table 3.1. Table of driver activity classes and class distributions.

The goal is to correctly and easily classify the given classes. More commonly, tasks similar to this are approached by training or fine-tuning vision classifiers, such as Convolutional Neural Networks (CNNs) or Vision Transformers. Ultimately, utilizing a model with video understanding is the more logical solution. The best performing model from the corresponding 2023 challenge utilizes a Video Transformer per camera view and utilizes the embeddings from those to train a seperate network [23] acchieving a validation accuracy of approximately 76.6 % (utilizing only five-folds). This however, as observed in Chapter 4 is a costly process due to the usage of Video Transformers. This, as well as a model perfectly trained and fitted to a specific situation would require adjustments if new classes or functionalities were introduced.

This work explores the application of generalizable Vision-Language models to classify these activities without extensive tuning utilizing single frames, emphasizing the use of pre-trained models to predict in-car activities based solely on visual inputs.

3.1 Vision Language Models: Bridging Visual and Linguistic Representation

Vision-Language models represent a significant advancement in the field of artificial intelligence, where the convergence of visual and linguistic data processing leads to a more integrated understanding of multimodal inputs [24, 25]. These models are designed to comprehend and generate responses based on a combination of visual cues and textual data, facilitating numerous applications such as image captioning and visual question answering.

This dual-input capability is typically achieved through the use of neural networks that process images and text separately but in parallel. The image processing component usually employs Convolutional Neural Networks (CNNs) or Vision Transformers. On the other hand, the textual component often utilizes Transformers to handle linguistic encoding.

3.1.1 CLIP: Contrastive Language–Image Pre-training

CLIP (Contrastive Language–Image Pre-training) represents a significant leap in Vision-Language models, offering a novel approach to learning visual concepts directly from natural language descriptions. Developed by OpenAI, CLIP offers a bridge for the gap between visual data and textual information by learning to associate images with captions. This model is capable of understanding and predicting complex visual concepts in a zero-shot framework, meaning it can classify images it has never seen during training based solely on textual descriptions, making it relevant for this experiment. [24]

Architecture and Pre-training

CLIP is dual-structured, comprising two primary components: an image encoder and a text encoder. The utilized image encoder is Vision Transformer [26], while the text encoder utilizes a Transformer architecture. These encoders transform their respective inputs—images and text—into a shared embedding space, with an example as in Figure 3.2.

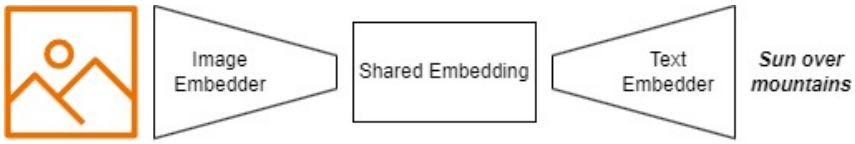


Figure 3.2. Example of the shared CLIP embedding, representing the similarities between the image and text inputs.

The core innovation in CLIP lies in its training methodology, which employs a contrastive learning objective. The fundamental idea behind contrastive learning is to learn an embedding space where similar items are brought closer together and dissimilar items are pushed apart [27]. The usecase for CLIP is however, as illustrated in Figure 3.3, that the diagonal, representing the similarity, is being maximized and the rest minimized. This, as the similarities are achieved by calculating the cosine-similarity for the given embeddings, and maximizing the cosine value of

the caption and image that should fit together, the diagonal, while minimizing the others, as the most similar cosine-similarity score is 1.

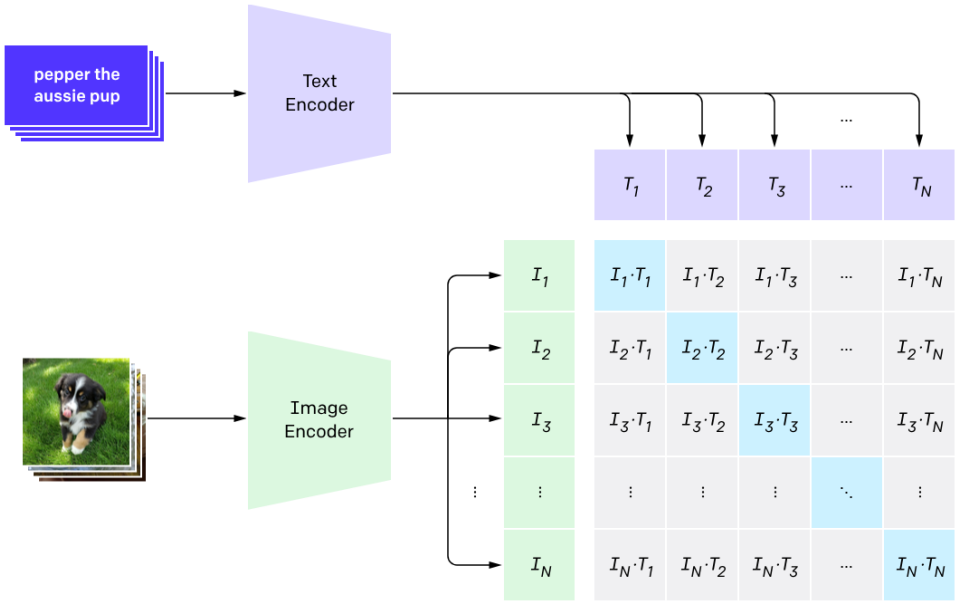


Figure 3.3. Example of the contrastive image-language training, resulting in shared embeddings as a result of the optimization of good cosine similarity pairs of image embeddings (I) and text embeddings (T) [24].

During pre-training, CLIP is exposed to a vast dataset of 400 million image-text pairs gathered from the internet. It learns by predicting the correct pairing of text and images among a batch of candidates, generating the closest image-text embeddings.

Zero-Shot Capabilities

One of the most interesting capabilities of CLIP is its performance in zero-shot scenarios. After pre-training, CLIP can directly apply learned visual concepts to new tasks without further training. For instance, without any specific tuning, CLIP performs similarly for the classification accuracies of ResNet-50 on ImageNet, all in a zero-shot manner. This is done by leveraging the embeddings from the pre-trained model to classify new images based on text descriptions alone, demonstrating a broad understanding of visual content that is significantly generalizable across different domains. Following the previous training example in Figure 3.3, the trained model can be utilized as in Figure 3.4, removing the requirements for fine-tuning a classifier.

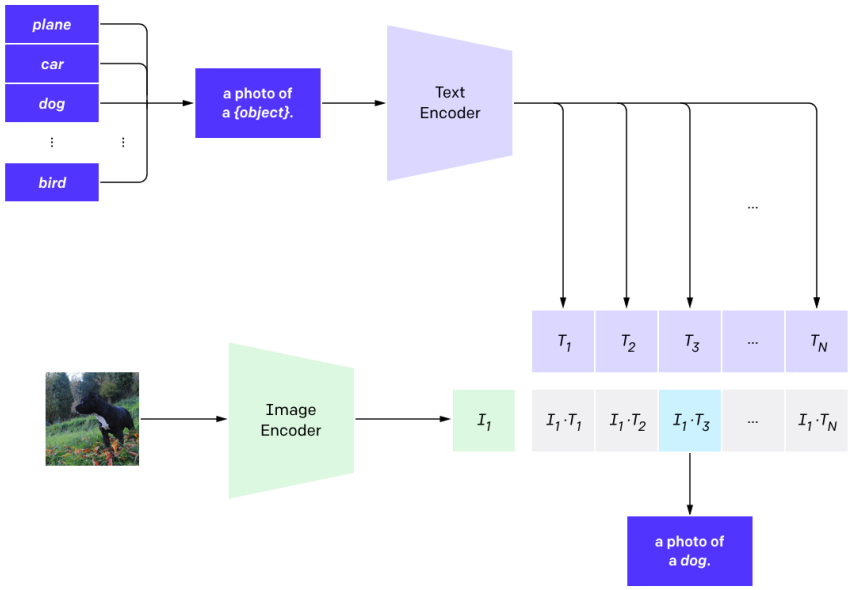


Figure 3.4. The CLIP zero-shot approach, utilizing the cosine similarity between the image embeddings (I) and text embeddings (T) [24].

3.2 Experiment

We want to investigate whether the CLIP visual embeddings can be utilized for a multi-view classification task. The dataset employed in this study features three distinct views, necessitating a classification approach that diverges from the one described in the previous section. This adjustment is crucial since no single view can accurately classify all classes of the given dataset; some classes are identifiable in one view but not in others, and some are visible across all views. For instance, the scenario of holding a phone in the left hand, class 3 *Phone Call (left)*, depicted in Figure 3.5, exemplifies a case where accurate classification is possible only by leveraging data from two of the three cameras. Some cases such as class 8, *adjust control panel*, is often only classifiable from the side-view camera (3), as the subject is often still focusing on the road and the upper arm and shoulder area may not indicate any change.

Figure 3.5. Illustration of multi-perspective in-cabin camera views for monitoring driver behavior under the class '0: Normal Forward Driving'. (1) Dashboard view. (2) Rear-view. (3) Side view. The phone is visible from angle 0 and 1 while being difficult to detect from angle 2.

To make up for the multiple views, the experiment employs a three-view approach, leveraging vision embeddings predicted from the pre-trained CLIP Vision Transformer model[1]. These

[1]The generated image embeddings are made available at https://kaggle.com/mathiasviborg/multiview-CLIP-generated-embeddings

embeddings are processed by three distinct neural networks, which are fused into a single network for the final prediction, as depicted in Figure 3.6. This integrated system is designed to identify key features within the embeddings that are crucial for classifying specific categories and take advantage of the different camera views, potentially learning to put importance to the embeddings from camera 1 and 2 when classifying class 3.

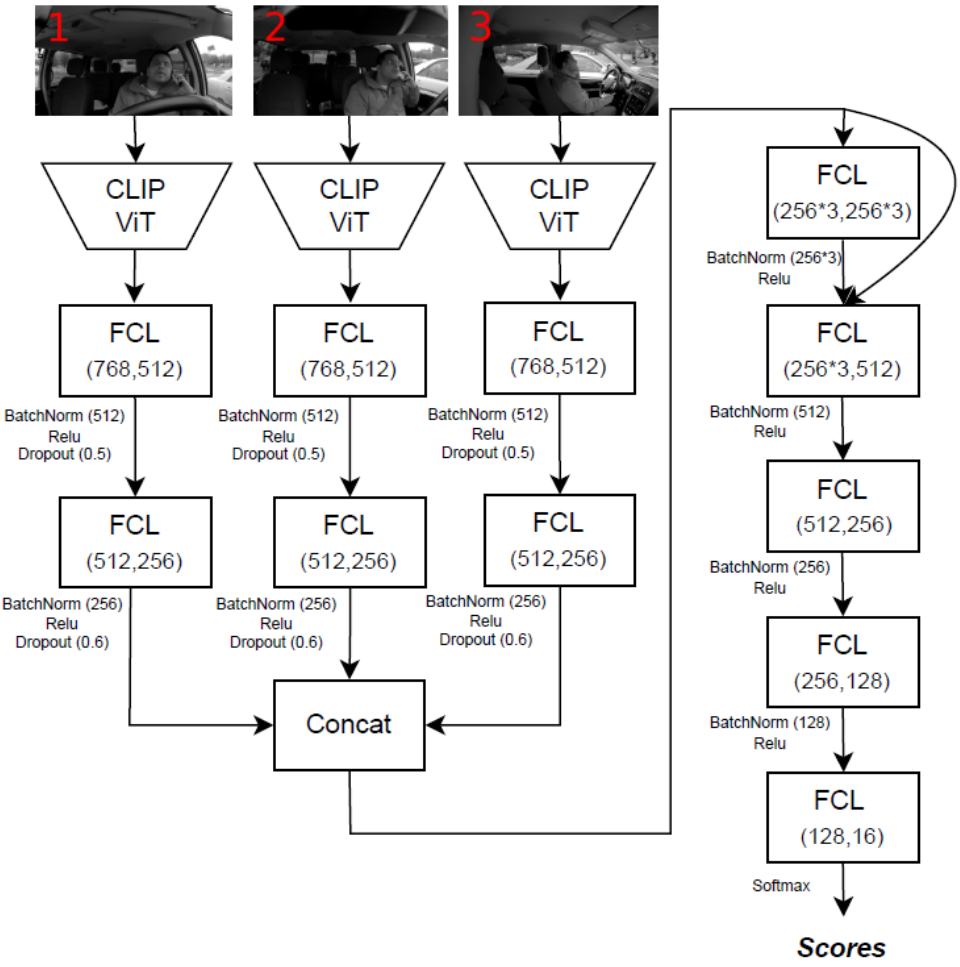


Figure 3.6. Experimentally chosen architecture, Semantic Representation Late Fusion Neural Network (SRLF), of 2.7 million trainable parameters utilizing embeddings from the CLIP Image Encoder. Utilized clip embeddings are of length 768. The order of the input images are randomized in order to combat overfitting and make generalization better.

The dataset is divided into 80 % for training and 20 % for validation, with a substantial batch size of 640 to balance computational efficiency, memory utilization and generalization. The model underwent 100 epochs of training, employing the cross-entropy loss function suited for multi-class classification tasks. The Adam optimizer is utilized with an initial learning rate of 0.0001. To mitigate overfitting, dropout rates of 0.5 and 0.6 were applied alongside batch normalization at each linear layer to stabilize learning as well as early stopping with a patience of three epochs.

Utilizing this approach alone results in an average accuracy of 71.64 %, utilizing a k-fold validation strategy as in Table 3.2.

K-fold	Accuracy
1	68.09 %
2	74.40 %
3	73.60 %
4	71.37 %
5	70.15 %
6	75.34 %
7	68.53 %
Average:	**71.64**
Std.:	**2.88**

Table 3.2. Table of k-fold cross-validation accuracies, average accuracy and standard deviation.

However, given the sequential characteristics of the dataset, it is possible to enhance these results by integrating filtering methods. Specifically, the dataset annotations designate numerous continuous seconds to a single class, indicating that all frames within this interval should belong to the same class. As the current methodology validates each frame independently, incorporating filtering techniques into the system could improve accuracies by removing outlier predictions.

To effectively address the outliers within the predictions, a Mode Filter is implemented. This filter operates by sliding a specified-sized window across the predictions. Within each window, the Mode Filter identifies the most frequently occurring value (mode) and assigns it to the central data point. This method helps to adjust isolated predictions that do not conform to their surrounding data points, thereby enhancing consistency and reducing noise [28].

Before:	1	1	0	1	0	1	1
After:	1	1	1	1	1	1	1

Table 3.3. Illustration of the Mode Filter effect on given example data with a window size W=5.

The window size is a critical parameter and must be an odd integer to maintain symmetry around the central data point. Smaller window sizes retain more granularity but may be less effective at filtering out noise, whereas larger windows provide greater smoothing at the risk of oversimplifying the data. For this case, a window size of 141 has been experimentally chosen as in Figure 3.7, corresponding to 4.7 seconds at an fps value of 30.

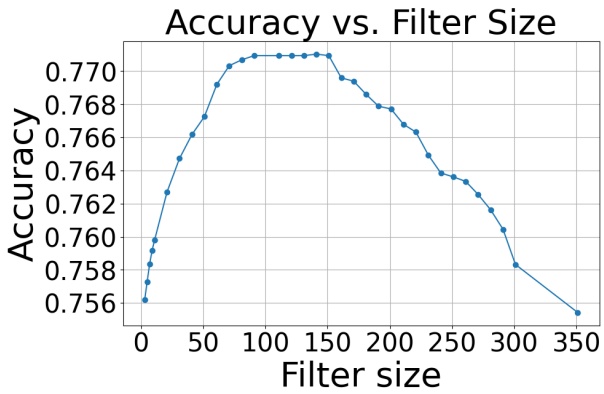


Figure 3.7. Example of experimentally choosing filter size for the best performing Fold 6.

Correspondingly, the average duration of a task (excluding class 0) is 11.77 seconds with a min of 5.09 seconds (class 5) and a max of 18.00 seconds (class 6). Implementing this method on the best-performing Fold 6 yielded an enhanced accuracy of 77.10 %, and yields updated accuracies as in Table 3.4.

K-fold	Accuracy
1	70.02 %
2	75.32 %
3	74.98 %
4	73.12 %
5	72.64 %
6	77.10 %
7	70.83 %
Average:	**73.43**
Std:	**2.35**

Table 3.4. Table of k-fold cross-validation accuracies including filtering and average accuracy.

Upon examining the distribution of predictions, as illustrated in Figure 3.8, it is evident that the predictions are biased among the classes. This skewness is attributed to the uneven distribution of data, as detailed in Table 3.1.

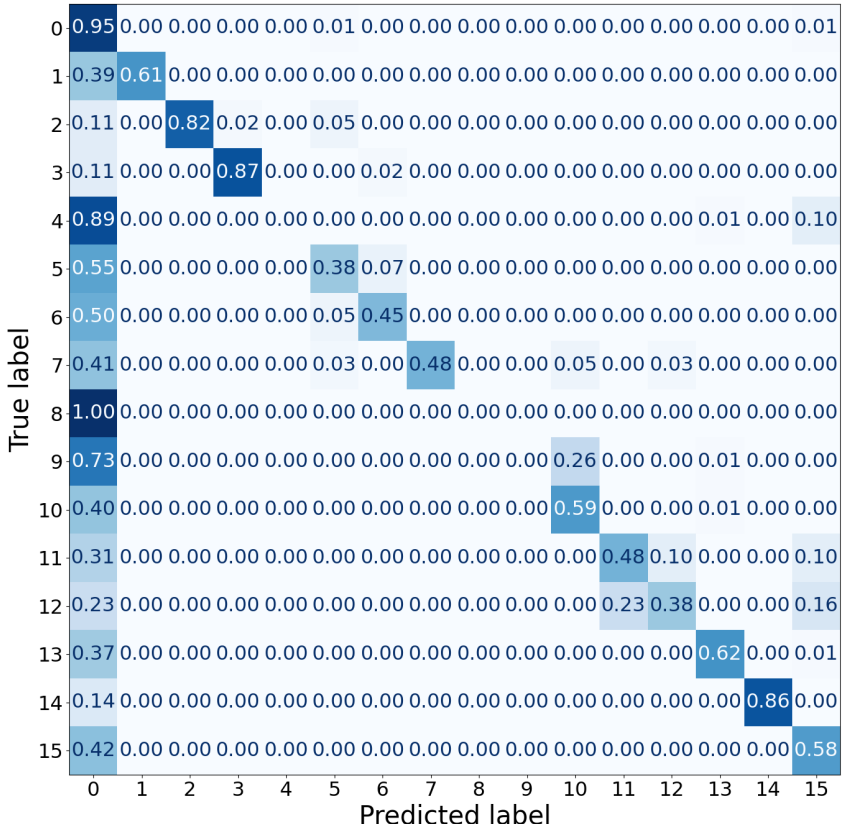


Figure 3.8. Confusion matrix for best performing k-fold 6 including a mode filter, resulting in a performance of 77.10 %. The same pattern is observed over the other k-folds.

Attempts to address this skewness have included various methods such as class weights, weight

decay, adjustments to dropout rates and numerous network architectures. Nevertheless, while retaining all classes, the most effective solution has been to load every 20th sample from class 0. This approach led to a testing accuracy of 55.19 % after filtering (51.63 % before filtering), still testing on all test data with no reduction. Since class 0 represents only 6.57 % of the data, this strategy marks a significant improvement in model generalization, as illustrated in Figure 3.9.

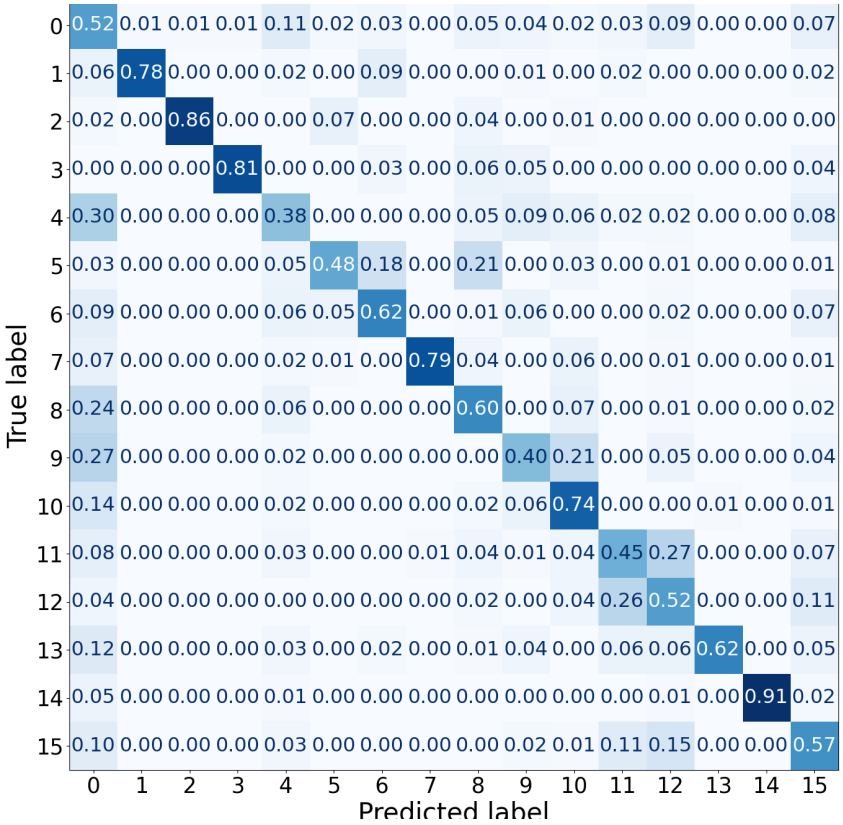


Figure 3.9. Confusion matrix for best performing k-fold 6 including a mode filter, resulting in a performance of 55.19 % after reducing the overall train sample count of class 0 to every 20th sample.

As the model is still struggling to generalize all classes, simply removing class 0 is investigated with results in Figure 3.11. Class 0 is removed as the straightforward driving class is the class counting the most false-negatives, leading to confusion and is struggling even in a binary use case as shown in Figure 3.10.

The generalization of the model in Figure 3.11 is considerably enhanced, still achieving an accuracy of 70.06 % which still is lower than the first achieved 77.10 % where class 0 was heavily over sampled. It is important to note that the distribution across the classes is more balanced, leading us to believe that generalizable system is useful.

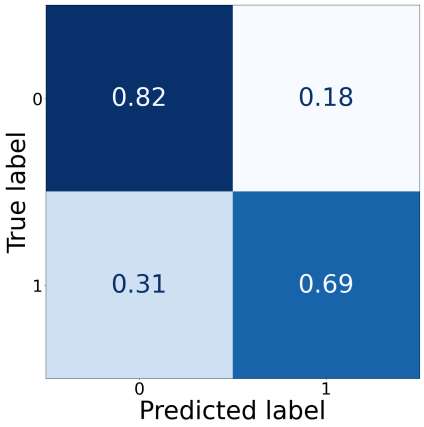


Figure 3.10. Binary Confusion matrix for best performing k-fold 6 only including class 0 for straight forward driving and a combination of all other activity classes, performing 77.22 % accuracy.

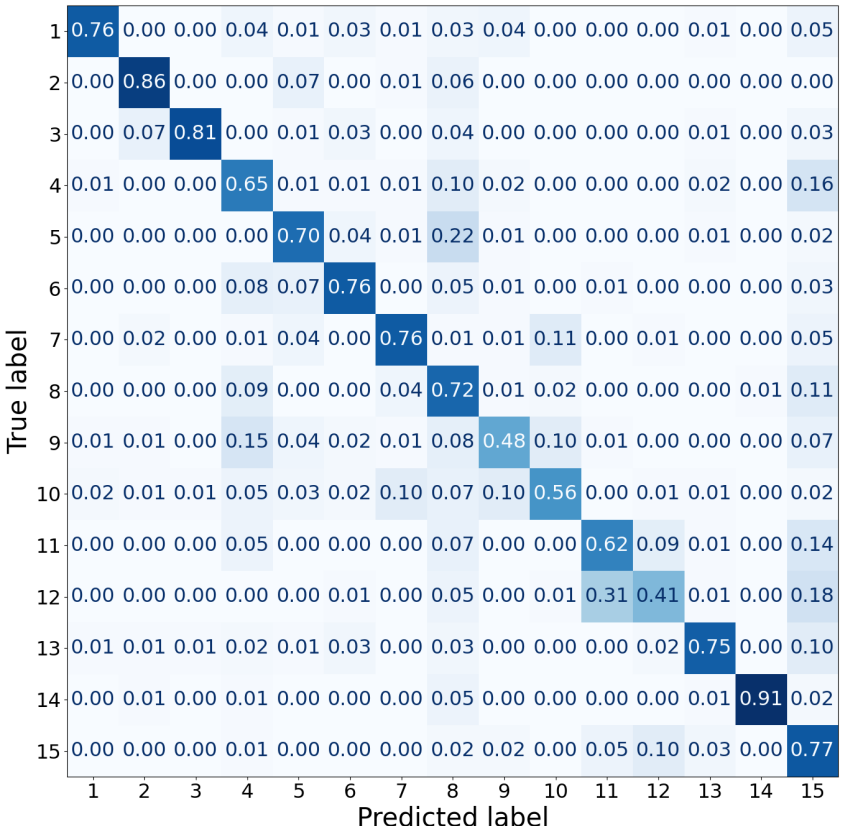


Figure 3.11. Confusion matrix for best performing k-fold 6 without class 0 for straight forward driving and including a mode filter, performing 70.06% accuracy.

However, CLIP encounters difficulties with classes 9, 10, 11, and 12, which involve specific actions in particular locations. For instance, classes 9 and 10 involve *picking up from the floor* on the driver and passenger sides, respectively, while classes 11 and 12 involve *talking to a passenger* on the right and in the backseat, respectively. A general encounter for this experiment is that these classes always seem to perform in the low end, or mostly predict one over the other, indicating a struggle for the general model as it now has to describe two things at once.

3.3 Concluding remarks

The results of this experiment display the effectiveness of Vision-Language models, as the produced CLIP pre-trained visual embeddings are enough to train a low-cost network of 2.7 million parameters, skipping the process of potentially fine-tuning Vision Transformer ($\approx$ 90 million) for the classification task. A well-generalizing accuracy of 70.06 % is reached without class 0, showcasing the promising, but still developing field of generalized Vision-Language classification. The overall model however struggles to differentiate between class 0 and other actions. Had the dataset been captured in a real setting, it's anticipated that the model would yield better performance, as the model would be allowed more visual cues. This, as classes such as talking to a passenger on the backseat currently contain an act of speaking to a fictive person on the backseat, eating involves an act of movement with no real food present, and driving straight forward involves just looking in the forward direction in the car, with no further actions

such as turning the wheel.

Future avenues for this research would be to investigate these principles within more video related areas such as Video-Language models as well as investigating other vision based approaches [29]. The Vision-Language model LLaVA 13b has been investigated, offerign a different language approach in contrast of CLIP, and it does not indicate any better results nor is it more efficient in handling the data. A general observation is that it struggles greatly with predicting left and right as well as predicting if an action is happening in the passenger area, back seat, and similar, which is similar to the conducted CLIP experiment. This indicates that the generalizable models may still have general struggles in the areas of a deeper understanding of simple classifications. This even as the general idea of utilizing multi-view is to utilize the knowledge of an action not being visible from one angle, potentially indicating left or right. LLaVA 70b might yield better results, and fine-tuning may be necessary.

During model testing architectures such as LSTMs and Transformers for processing the CLIP image embeddings have been investigated, not yielding better results. It was hypothesized that the temporal nature would potentially learn the filtering, which proved to not be the case. Given a larger amount of data, these methods are believed to still hold promise.

Given its ability to detect yawning (class 13), it would also be intriguing to conduct a similar experiment to classify drowsiness using the dataset presented in Chapter 4 and investigate whether a generalizable model will be able to identify drowsiness features.

Drowsiness Detection Utilizing Video Transformers 4

This section aims to examine the use of Video Transformers for drowsiness detection, specifically focusing on the necessary temporal detail in videos for precise classification and the optimal video duration.

The research presented here has been conducted in a worksheet format, and no formal paper has been written as a result of this content.

The processed dataset in this section can be found at https://www.kaggle.com/mathiasviborg/uta-rldd-videos-cropped-by-faces.

4.1 Is drowsiness a problem?

Drowsy driving is a significant yet often underappreciated problem that poses serious risks to road safety. Statistics indicate that fatigue contributes to 10-25 % of all road accidents, emphasizing the widespread nature of this issue [30]. According to the National Highway Traffic Safety Administration, in 2017 alone, drowsy driving was responsible for approximately 91,000 crashes in the United States (US), resulting in around 50,000 injuries and nearly 800 deaths [31]. Despite these concerning statistics, the real impact of drowsy driving is believed to be even greater, as reported by experts from fields such as traffic safety, sleep science, and public health. Furthermore, a study by the National Sleep Foundation revealed that 54 % of drivers admitted to feeling drowsy while driving, and 28 % confessed to having fallen asleep at the wheel at least once [32].

Given the severe consequences of drowsy driving, developing reliable detection systems is not just a technological challenge but a critical societal need. Many countries, including those in the European Union (EU) and the US, have taken steps and implemented regulations that mandate rest periods for commercial drivers to mitigate this risk as well as regulating driving hours [33, 34]. Some states within the US, such as New Jersey, actively mention drowsy driving as reckless [35]. The EU is expanding on the laws for the area demanding that most vehicles utilized for passenger and goods transport sold after 7th of July 2024 must have a Driver Drowsiness and Attention Warning (DDAW) system [36], indicating the recognition of the problem as a part of a goal to cut road fatalities and injuries in half by 2030 [37].

Research has shown that the effects of drowsy driving can be compared to driving under the influence of alcohol. For instance, a study illustrated in the following Table 4.1 shows the blood alcohol content (BAC) equivalent for various durations of wakefulness:

Hours Awake	BAC [g/liter]
24	0.1
20	0.08
18	0.05

Table 4.1. Corresponding Blood Alcohol Content (BAC) levels to hours awake when driving without adequate sleep [38].

Considering the breadth and importance of alcohol-driving-related legislation across the EU, the US and globally, it's essential to address not only alcohol impairment, but also the strong correlation for driver fatigue. In the US, the widespread Blood Alcohol Concentration (BAC) limit is set at 0.08 to prevent alcohol-impaired driving [39]. Similarly, other countries have taken measures to ensure road safety. For instance, many European nations, such as Denmark, enforce a lower BAC threshold of 0.05, and the Czech Republic even has adopted a BAC threshold of 0.0 [40]. Alcohol-impaired driving has been studied for decades, with a 1996 study highlighting the risks associated with even minor increases in BAC:

> ***"Each 0.02 percent increase in BAC nearly doubles a driver's risk of being in a fatal crash."*** [41]

This knowledge underscores the necessity of stringent policies to mitigate all forms of driver impairment, directly translating to measurements for drowsiness driving, ensuring safer roads for everyone. One challenge, however, is that while alcohol levels can be measured using breathalyzers and similar devices, detecting drowsiness requires different approaches. Unlike alcohol, which has a quantifiable presence in the bloodstream that can be tested objectively, drowsiness does not have a direct physical marker that can be similarly measured. It however can be measured using biological-based methods such as brain signal-based methods and eye signal-based methods [42]. This necessitates the use of alternative methods to assess fatigue and ensure safety. Advancements in sensor technologies and machine learning algorithms are proving effective in identifying signs of fatigue in drivers in real-time and is used in many of today's cars [32, 43]. To enhance the understanding of drowsiness detection, the objective of this study is to investigate the use of temporal Transformers in the realm of identifying signs of driver fatigue and uncover potential new insight into the area.

4.1.1 Indicators of drowsiness

Drowsiness is the result of a biological need to sleep and happens for numerous reasons. Common causes include being awake for long periods, poor sleep quality, medication and similar factors [42]. Some visible clues often include the following [42, 44, 45, 46]:

- **Struggle to keep eyes open:** Often exhibiting frequent blinking or droopy eyelids. These signs are evident as the person battles the urge to sleep.
- **Difficulty in maintaining head posture**: A drowsy person's head may nod forward or sway involuntarily, signaling a loss of muscle control as the body succumbs to sleep.
- **Frequent yawning:** Clear sign that the body is craving rest.

- **Microsleeps:** These brief, involuntary episodes of sleep last from a mere fraction of a second up to 30 seconds. During microsleeps, a sudden lapse in attention occurs, often unbeknownst to the individual, which poses substantial risks during tasks like driving.
- **Rubbing Eyes:** Tired individuals often rub their eyes to stimulate them and temporarily reduce the effects of fatigue.
- **Breathing Patterns:** As drowsiness intensifies, the frequency and depth of breathing may change; breaths tend to slow down but become deeper when falling asleep.
- **Physical Appearance:** Increased under-eye bags, a generally tired look, and diminished frequency in smiling or engaging in conversation can also indicate fatigue.

Other indicators can be such as unjustifiable variations in speed, slowly drifting out of a lane while driving, difficulty concentrating and slower decision/reaction making. The signs will become more and more visible the more drowsy a person becomes, with levels specified as in Table 4.2, following the Karolinska Sleepyness Scale (KSS).

1	Extremely alert
2	Very alert
3	Alert
4	Fairly alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep alert
8	Sleepy, some effort to keep alert
9	Very sleepy, great effort to keep alert, fighting sleep

Table 4.2. The Karolinska Sleepyness Scale (KSS) version B categories. Often used as 1-3 (blue) as alertness, 6-7 (orange) as low vigilance, and 8-9 (red) as drowsy [47][48]. Further use in this work will utilize the blue and red areas as a binary classification task.

4.2 Related works

Drowsiness detection is a well-studied area within the field of automated safety systems, particularly utilizing visual cues from drivers or operators to signal fatigue early.

Monitoring technologies have already been implemented in several car models. Volvo pioneered this initiative in 2007 with the introduction of their attention system [42]. Following Volvo's lead, other major automotive brands such as Mercedes [49], Volkswagen [50], BMW [51], and Ford [52] have developed and integrated similar systems into their vehicles followed by many others. Commonly an eye tracking system is utilized, while also taking account for other data modalities such as the way a given driver is rotating the steering wheel of the car, as the Bosch system in Figure 4.1.

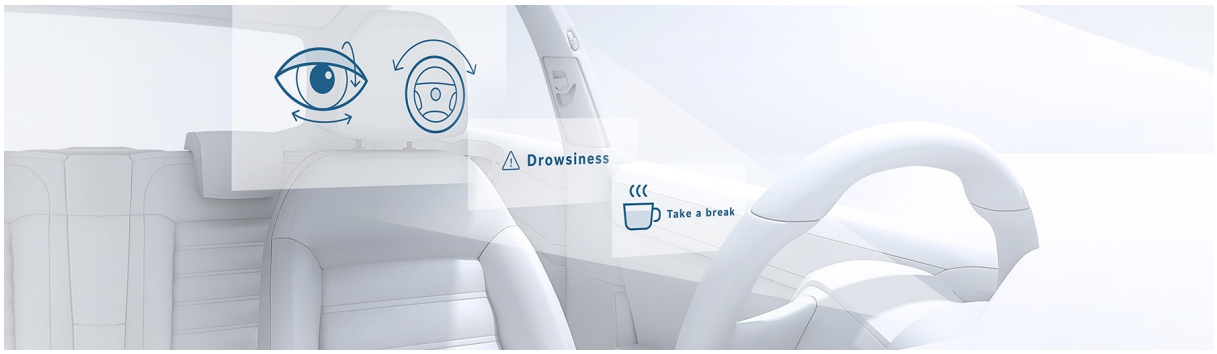


Figure 4.1. Illustration of Bosch's detection system designed for integration into automobiles [43].

Recent public advancements have leveraged deep learning techniques to enhance detection accuracy and adaptability across different scenarios commonly using publicly available datasets such at the University of Texas Arlington Real-Life Drowsiness Dataset (UTA-RLDD), which will be utilized in this report, at approximately 30 hours of data [53] and the Yaw Drowsiness Dataset (YawDD) at approximately 2.5 hours[1] [54], while often validating on own acquired test data [55].

The authors behind the UTA-RLDD dataset capitalize on the sequential arrangement of frames, analyzing blinking patterns as a means to detect drowsiness. They extract blink sequences which are then input into a temporal LSTM model specifically designed to identify signs of drowsiness by examining the timing and frequency of eye blinks. This network is designed to leverage the temporal pattern in blinking, acknowledging that the relationship between blinks and their succession can influence drowsiness detection, achieving an accuracy of 65.2 %, improving their compared 57.8 % human judgement score.

More recent advancements in model architectures include utilizing CNNs [56] and Vision Transformer [57] looking at single frames and utilizing only facial cropping as preprocessing. These have resulted in binary accuracies of 91 % on both training and validating on UTA-RLDD, significantly outperforming the original 65.2 %.

Even as the current research indicates that a single frame can classify drowsiness well, it is intriguing to explore how similar techniques such as Video Transformers might perform when incorporating temporal information on the same dataset as the intuition is to achieve an even higher accuracy.

4.3 Dataset

For the purpose of detecting drowsiness, a suitable dataset is needed. In this study, the University of Texas at Arlington Real-Life Drowsiness Dataset (UTA-RLDD) will be utilized [47]. The UTA-RLDD is unique in its focus on multi-stage drowsiness detection, capturing not only extreme and easily visible cases of drowsiness but also subtle cases where micro-expressions are the discriminative factors.

[1] Number achieved by analyzing dataset utilizing OpenCV.

The UTA-RLDD comprises approximately 30 hours of video content, showcasing a range of drowsiness levels from subtle signs to more obvious ones. It includes RGB videos of 60 healthy participants, with each participant contributing one video for each of three different classes: alertness, low vigilance, and drowsiness, resulting in a total of 180 videos. The participants record the data and are told to do so when they feel very awake, signs of sleepyness appear, and on the verge of falling asleep corresponding to blue, orange and red on the KSS. The dataset follows the Karolinska Sleepiness Scale (KSS) version B [48], which can be identified in Table 4.2.

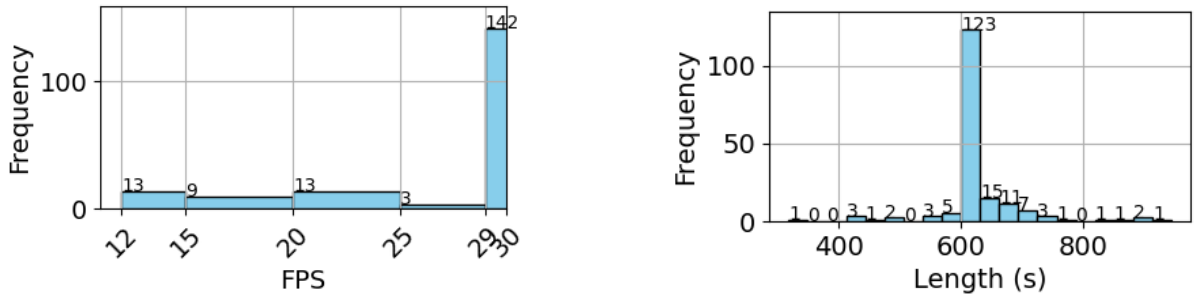
The participants are a diverse group, all over 18 years old. The group included 51 men and 9 women of various ethnicities and ages. Some of the videos feature subjects with glasses (21 out of 180 videos) and considerable facial hair (72 out of 180 videos), adding to the diversity and real-world applicability of the dataset with examples showcased in Figure 4.2. Each video is self-recorded by the participant using their cell phone or web camera, representative of the frame rate expected of typical cameras used by the general population.

Figure 4.2. Displayed are frames extracted from the UTA-RLDD dataset, representing different states: alertness (top row), low vigilance (middle row), and drowsiness (bottom row) [53].

The UTA-RLDD is at the time of release, and to the knowledge of the author, the largest realistic drowsiness dataset freely available.

4.3.1 Preprocessing

As each video is self-recorded, and different cameras have different standard settings and build, a large number of the videos are of different resolution (Figure B.1 in Appendix B), frame rate (Figure 4.3a), length (Figure 4.3b), file format, frequently contain shaking web- and phone-cameras and general instability in the recordings. To fulfill the objectives of this study, which focus on analyzing facial expressions and movements, it is wanted to isolate and crop out these specific features from the videos.



(a) (b)

Figure 4.3. Histograms illustrating the distribution of frames per second (fps) (a) and the length of the videos (b) in the dataset.

A facial-cropped per frame version of this dataset already exists, with one example being the Driver Drowsiness Dataset (DDD) [58]. This however does not contain all subjects, nor all frames from the subjects it contains. Experimentally making videos from the given frames results in a frame rate per second of 5, whereas the minimum of UTA-RLDD is 12 fps (4.3a). More so, to use the frames in a sequence purpose, the cropping must be somewhat identical in every frame to obtain stabilization, otherwise the movement potentially can be misunderstood by a temporal classifier. When observing the data, all crops are of an unstable size and therefore unfit for a smooth video. A dataset therefore is extracted manually.

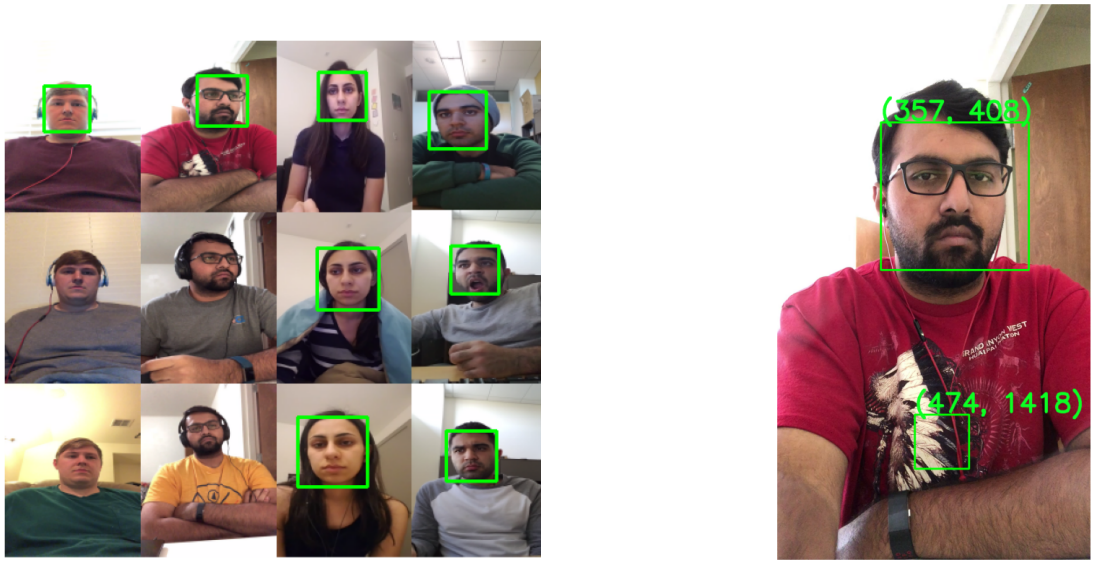
Cropping

Cropping plays a pivotal role in isolating and extracting facial expressions and movements from videos, which is essential for the objectives of this study. However, selecting an appropriate facial detection method is crucial, considering the variability in video resolution, frame rate, and camera stability across the dataset.

Despite the availability of multiple tools, the primary considerations for this project are the accuracy of prediction and processing speed due to the large amount of facial detection needed, as at 10 fps the dataset would yield approximately 720 thousand frames.

Initially, Haar Cascades (Viola-Jones algorithm) [59, 60] was considered for its simplicity and speed as a more traditional light weight machine learning-based classifier commonly used for object detection, including faces. It operates by scanning an image with a sliding window technique, applying pre-defined Haar-like features to each window and using them to classify whether the region contains the object of interest, such as a face. It employs a cascaded classifier

architecture comprising multiple stages of classification. Each stage applies a classifier to a region of the image, and if the region passes the classification at a particular stage, it progresses to the next stage for further refinement. This cascaded structure enables the algorithm to efficiently discard non-object regions early in the process, reducing the computational burden on subsequent stages. However, when experimenting, the performance has proven to be inconsistent, often detecting faces inaccurately or missing them entirely, particularly in instances of non-standard facial positions. This inconsistency was clear when observing predictions as in Figure 4.4a, and some cases were even predictions not containing a face as in Figure 4.4b. The speed, however, is unmatched, averaging 0.06 seconds over 12 instances in the example below.



(a) (b)

Figure 4.4. Evaluation of the OpenCV [61] implementation of Haar Cascades for facial detection, revealing its proficiency in predicting certain facial features while demonstrating limitations in recognizing others.

A more recept option using deep neural networks for facial detection is a single-stage model such as RetinaFace [62][63] which is known for yielding better performance in faces across diverse poses, lighting conditions and scales. It utilizes optimized anchor boxes, instead of a sliding window, within a pyramid structure to efficiently capture faces at different scales and aspect ratios across the image. These anchor boxes are strategically placed at various levels of the pyramid, ensuring comprehensive coverage of the image space and enabling precise localization of faces. However, its computational complexity may lead to slower processing speeds, especially when handling extensive video datasets. Figure 4.5 displays an example of a well performing RetinaFace, with a slightly slower processing time.

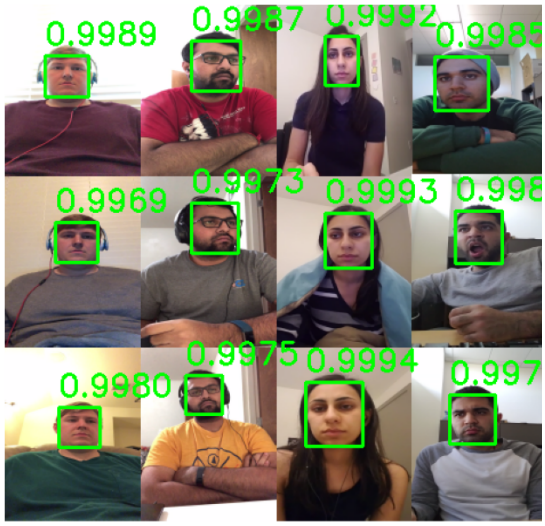


Figure 4.5. Prediction evaluation of RetinaFace, revealing its proficiency consistent predictions. The average speed over 10 runs is 0.14 seconds.

A speed assessment on 100 individual frames is conducted to accurately measure the time for individual frame prediction only containing one face. The results of this process for each of the three detectors are outlined in Table 4.3.

Detector	Time (s)	Std.
Haar Cascades	0.100	0.019
RetinaFace	0.151	0.009

Table 4.3. Average time of 100 predictions in seconds. RetinaFace utilize GPU whereas the Haar Cascades implementation utilize CPU.

The overall test results indicate RetinaFace as the most beneficial option, as it's speed on the given setup is comparable to the one of the Viola-Jones algorithm while outperforming the facial predictions.

Smoothing

In the process of creating a video from these cropped segments, it's crucial to consider the inherent instability of both the original video content and the cropping process itself. The ideal scenario would be to have the same crop coordinates throughout the video. However, while some subjects may exhibit minimal movement throughout the recording, even subtle movements become highly noticeable in a video context. As depicted in Figure 4.6, an initial movement is observed as the subject's head is making movement, followed by a period of relative stillness. In this scenario, both the horizontal (x) and vertical (y) coordinates of the bounding box's upper left corner are adjusted, revealing visible fluctuations when predicting facial locations for each frame.

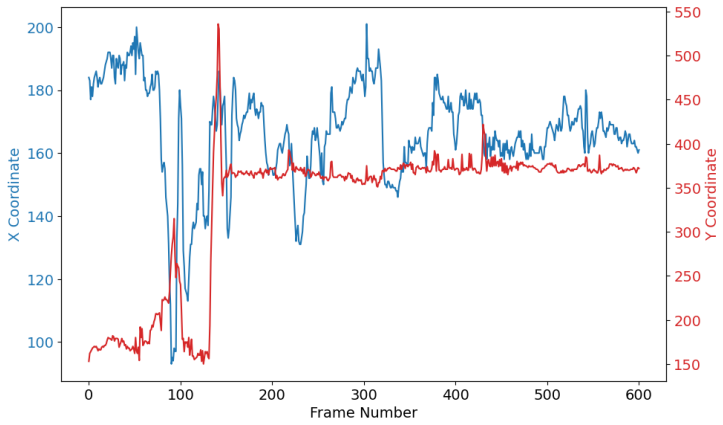


Figure 4.6. Unsmoothened top left corner (x,y) coordinate for crops in a videofeed. A significant subject movement occurs between frames 80 and 170, while the remainder of the frames remain mostly static. The first 60 seconds of subject 9 is used, at 10 fps.

When both observing the data in Figure 4.6 and the video it's clear that smoothing is needed. For the case of this data, the Exponentially Moving Average (EMA) [64] method is investigated (Equation 4.1) as a method of moving the bounding box on the image, to achieve a smoothed cropped video as the result.

$$y_n = (1 - \alpha) \cdot y_{n-1} + \alpha \cdot x_n \tag{4.1}$$

y_n	Smoothed coordinate.
α	Smoothing constant.
y_{n-1}	Previous smoothed coordinate.
x_n	Position of current coordinate.

The goal is to eliminate spikes in the signal, particularly when a new frame presents a predicted bounding box that may either be erroneous or result from minor, rapid movements of the subject that do not require adjustment. It is also common for the bounding box predictions to be unstable, even when the subject remains largely motionless. EMA implemented is seen in Figure 4.7.

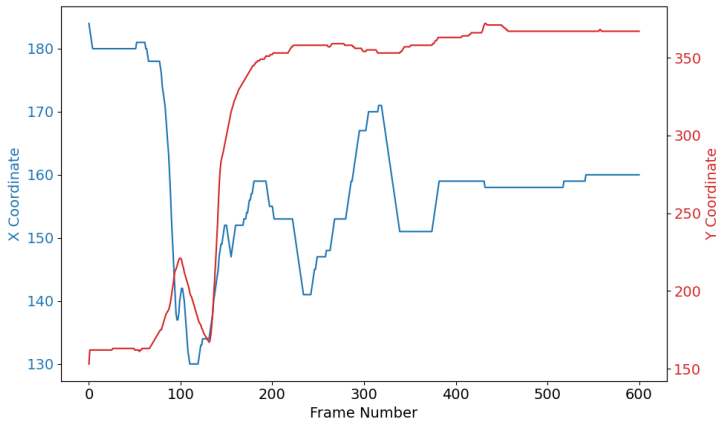


Figure 4.7. EMA, with $\alpha = 0.05$, smoothened top left corner (x,y) coordinate for crops in a videofeed. A significant subject movement occurs between frames 80 and 170, while the remainder of the frames remain mostly static. The first 60 seconds of subject 9 is used, at 10 fps.

To streamline processing times for the entire dataset, a strategy has been adopted to predict motion only for every 5th frame within a 10 fps configuration. This approach acknowledges that incremental movement between frames is typically insignificant. Furthermore, to capture a broader range of facial movement, an increase of 15 % in the bounding box size has been implemented, based on the larger dimension of either width or height, resulting in a difference as in Figure 4.8.

No margin. 15 % margin.

Figure 4.8. Example of of predicted bounding box before and after added margin. The added margin allows to detect more facial movement.

Moreover, a movement lock has been integrated to mitigate redundant frame adjustments, particularly when the newly predicted center lies within a distance less than 3 % of the frame's diagonal length. The inclusion of an extra margin facilitates this, affording the subject greater freedom of movement within the frame with the frame not needing to move as the subject is still captured fully. This results in a smooth almost non moving crop, as displayed in figure 4.9.

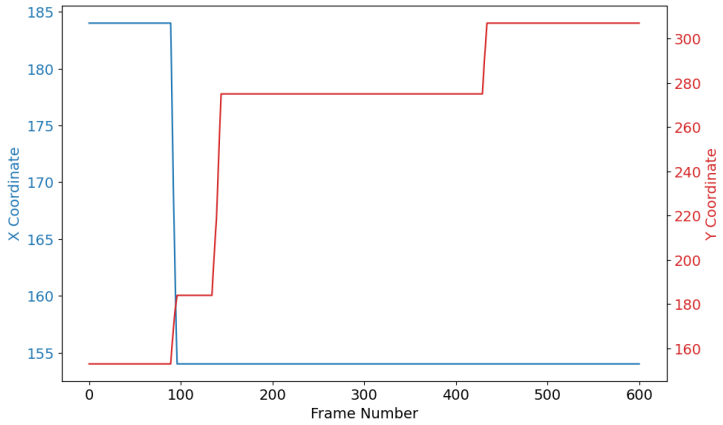


Figure 4.9. EMA, with $\alpha = 0.05$, smoothened top left corner (x,y) coordinate for crops in a videofeed. 3 % movement lock and predicting face from every 5th frame. A significant subject movement occurs between frames 80 and 170, while the remainder of the frames remain mostly static. The first 60 seconds of subject 9 is used, at 10 fps.

This approach is far more stable and avoids potential confusion for a neural network analyzing the movement of the crop while being more pleasant to watch. The initial and final 20 seconds of every video is disregarded because of significant visible motion, particularly focusing on adjusting cameras and similar.

The drowsy class of subject 42 has been dismissed as the videos contain large amounts of movement and camera shaking which the implemented tracking technique can not follow properly.

4.4 Temporal analysis strategy

In recent years, the field of natural language processing (NLP) has experienced a transformative shift with the advent of self-attention-based methods [65]. These methods, particularly the Transformer model, have demonstrated unparalleled success in capturing long-range dependencies between words, alongside offering scalable training solutions. As a result, they have established themselves as the standard across various NLP tasks such as question answering [66, 67].

Parallel to NLP, the domain of video understanding exhibits similair characteristics, primarily due to the sequential nature of both videos and textual content. In essence, just as the comprehension of a word relies on its context within a sentence, understanding momentary actions within videos often requires considering their broader context across the sequence. This similarity suggests that NLP's long-range self-attention models could be equally beneficial for video analysis, as Transformers has also proven valueble for single image classification, with architectures such as Vision Transformer [26].

Previously, the primary methodologies for video analysis have been dominated by 2D or 3D convolutions, focusing on spatiotemporal feature learning using architectures such as 3D CNNs [68] and hybrid approaches combining CNN and RNN architectures [69]. Newer models consist of Transformer architectures as TimeSformer [70].

4.4.1 Model overview

As the TimeSformer model allows for utilization of long duration clips, it has been selected as the Video Transformer architecture for this section. It introduces a novel approach to video analysis by leveraging the Transformer architecture, and adapting it to process video data. It does so by decomposing input video clips into a series of non-overlapping patches, treating each patch as a token similar to words in a sentence, and then applying self-attention mechanisms to capture both spatial and temporal relationships within the video. This methodology allows the TimeSformer to efficiently process and analyze video data, enabling competitive performance in action recognition and other video understanding tasks. [70]

Conceptually, it utilizes the same idea used for images as in Vision Transformer [26], and adapts it to video understanding, by upgrading the attention mechanism from image space attention, as in Figure 4.11, to space-time attention. Figure 4.10 displays the ViT model architecture adapted and expanded by TimeSformer.

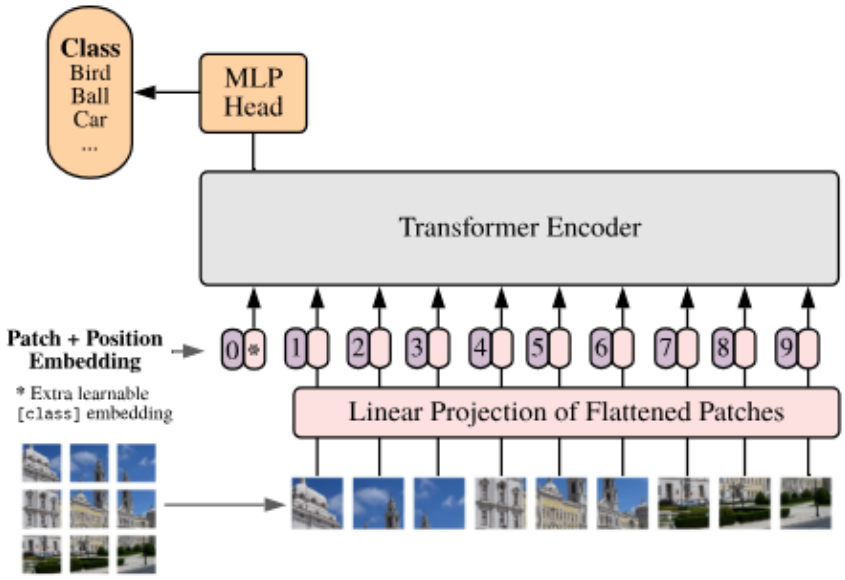


Figure 4.10. The Figure depicts the Vision Transformer (ViT) model architecture for image classification [26]. Images are segmented into patches and each patch is linearly projected. Alongside, a learnable class embedding is appended to the sequence of embedded patches. Positional embeddings are added to this sequence to maintain the positional information of each patch. The sequence is then fed to a standard Transformer encoder with self-attention mechanisms. Finally, the encoded features are passed to a Multilayer Perceptron (MLP) head, which outputs the classification results into predetermined categories such as bird, car, etc.

TimeSformer use the same patch model as ViT in Equation 4.2, decomposing the individual frames to no overlap patches.

$$N = H \cdot W / P^2 \tag{4.2}$$

N	Number of non overlapping patches.
$H \cdot W$	Height · Width.
P^2	Size of patch, each patch the size of $P \cdot P$.

The input of TimeSformer is wanted as visual RGB data in the dimension $\mathbb{R}^{H\times W\times 3\times F}$ with an introduced frame parameter (F). The patches are then flattened into vectors of the dimension $x_{(p,t)} \in \mathbb{R}^{3P^2}$, $p = 1, ...N$ being the spatial location within the frame and $t = 1, ...F$ being the time position of the frame. These embeddings are then linearly mapped to an embedding vector as Equation 4.3, utilizing a learnable matrix $E \in D^{D\times 3P^2}$, and including learnable elements with time position as an element of consideration, differing from ViT.

$$z^{(0)}_{(p,t)} = E \cdot x_{(p,t)} + e^{pos}_{(p,t)} \tag{4.3}$$

$z^{(0)}_{(p,t)} \in \mathbb{R}^D$	Embedded patch.
E	Learnable matrix used to embed the patches.
$x_{(p,t)}$	Flattened patch vectors including spatial and temporal imformation.
$e^{pos}_{(p,t)} \in \mathbb{R}^D$	Learnable positional embedding encodes each patch's spatiotemporal location.

In summary, an image is split into fixed-size patches, each of them linearly embeded, position embeddings added, and the resulting sequence of vectors is fed to a Transformer encoder, similar to ViT.

However, a significant difference hides in the attention mechanism. As multiple frames are used, focusing the attention on different frames is the goal, which is not fulfilled in the Space Attention used by ViT, see Figure 4.11. The TimeSformer paper investigates four different approaches, with the main being Divided Space-Time Attention (T+S), comparing a patch with the given frame while only comparing with the same patch location in other frames.

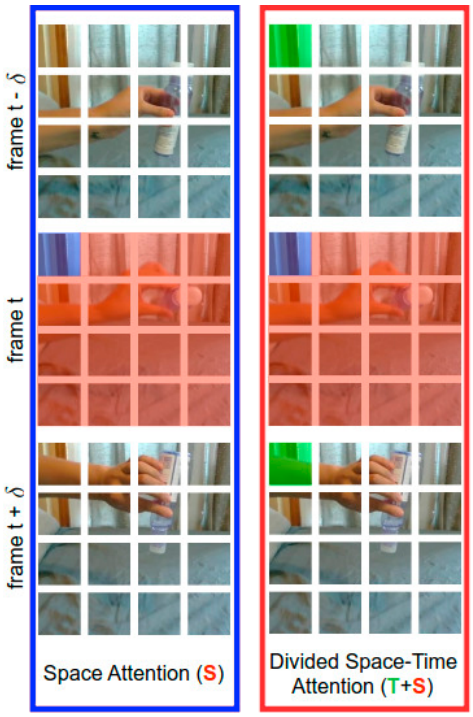


Figure 4.11. There are multiple ways to define a pixel's "neighborhood" when it comes to video data. A pixel's neighbor can be its spatial neighbors (blue) or spatial neighbors and temporal neighbors of only the location of interest (red). The Divided Space-Time Attention mechanism is utilized in the TimeSformer architecture [70]. Frame t represents the present frame, frame t - δ represents the previous frame, and frame t + δ represents the future frame.

TimeSformer ultimately uses Divided Space-Time Attentnion (T+S) as it has proven to create a better generalization and yield better performance accuracies on the Google Kinetics 400 dataset (K400) [71].

4.5 Sequential analysis

As a model and dataset now has been selected, an analysis can be performed. All analyses in this section will utilize the processed dataset described in Section 4.3, which comprises data from 60 subjects. A subset of these subjects will be reserved for testing to ensure the model is assessed on previously unseen data. The dataset consists of 5 folds, each consisting of 12 subjects.

A TimeSformer model pretrained on the Google Kinetics 400 (K400) dataset will be used and fine-tuned on the given dataset [70]. The K400 dataset is a human action video dataset, consisting of human behavior such as shaking hands, salsa dancing, etc. This dataset was chosen because it is the most closely related dataset for which a publicly available model exists. As transformers are known for requiring a large amount of data [72], training from scratch is not considered as that would require a much larger dataset, than the approximately 30 hours in the UTA dataset, as displayed in Table B.1 in Apprendix B. Figure 4.12 displays the significance of training data, as the accuracy improves by utilizing a larger amount of samples utilized from the K400 dataset.

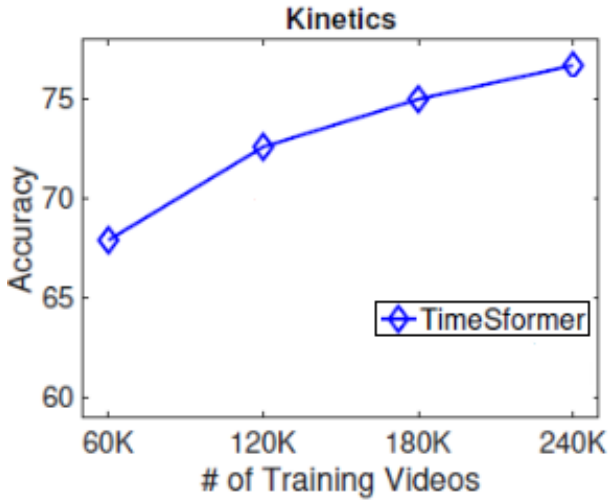


Figure 4.12. Accuracy on K400 when training on different subsets of the dataset [70]. It is observed that the performance is better the larger amount of training videos utilized from the dataset.

This study focuses solely on the *non-drowsy* and *drowsy* classes, corresponding to blue and red on the KSS, which provide approximately 20 hours of data. In comparison, the K400 dataset is approximately 850 hours, consisting of 400 classes each with a minimum 400 videos each of approximately 10 seconds [71].

4.5.1 Tuning TimeSformer for drowsiness detection

To enhance the effectiveness of the TimeSformer model in detecting drowsiness from video data, and discover which detail combination works best, it is crucial to determine optimal values for

various hyperparameters. The tool *Weights & Biases* is utilized for this purpose, as it provides capabilities for logging and analyzing data during the training of neural network models [73]. Besides storing historical runs, *Weights & Biases* offers hyperparameter sweeping tools to identify the best combinations of hyperparameters for the model.

Hyperparameter sweeping involves specifying a range of acceptable values for each hyperparameter and training multiple models with various combinations of these values. The most effective model is then selected based on performance. The hyperparameters adjusted in the sweep include video length and the number of frames from the video uniformly sampled by the model to investigate the sequential detail level needed for a good performance. Additionally, a skip factor for loading every nth data sample of the dataset, batch size and weight decay are included to prevent overfitting and improve generalizability. These adjustments are detailed in Table 4.4.

Hyperparameter	**Value [.]**
Video Length	5,10,20,30,60 [s]
Number of Frames	5,10,20,30,40,50,60,70,80,90,100
Skip-Selector	1,2,3,4,5,6,7,8,9,10
Batch Size	1,2,4,8,16,32
Weight Decay	min: 0, max: 4

Table 4.4. Overview of experimentally chosen range in hyperparameters and baseline values. Video length represent the length in seconds of a given video, number of frames represent the amount of frames uniformly sampled by the model from a given video and skip-selector represents sampling every nth sample of the dataset. A high value for weight decay has been chosen as early experimentation displays tendencies for overfitting and increased performance with a high weight-decay.

These parameters have been selected to explore whether drowsiness can be classified in a short or long duration of time as well and the amount of detail needed to be utilized by the TimeSformer model. Early testing has confirmed a pattern of overfitting, as displayed in Figure 4.13, resulting in the added variation of sweeping parameters to prevent overfitting and support a better generalization.

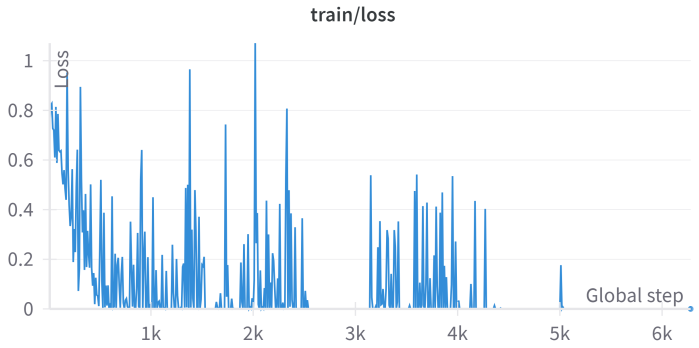


Figure 4.13. Selected run from initial testing without weight decay. This example has a batch-size of 2, utilizing 10 frames, video length of 20 and utilizes all data while fine-tuning all layers. 1 epoch = 1270 steps.

To further prevent overfitting, the example of Jaejun Lee et al. [74] is followed, ultimately freezing most layers and only fine-tuning a few, also improving training times for the sweep. For

this project, it's been chosen to only keep the last layer trainable, as it's experimentally proven to be the best performing. A model layer structure is shown in Table C.1 in Appendix C. The model architecture supports dropout, which has not been utilized in this experimentation as early experimentation showed limited improvements. This decision also enhances the precision of the sweeping process, as reducing the number of parameters decreases the amount of possible sweep combinations.

When sweeping the combinations chosen be be either of a random choice or simply trying everything in a grid-search. However, for a faster optimization a Bayesian Optimization technique has been applied. A Grid Search tests every possible combination within the specified values, whereas Random Search randomly selects combinations from these ranges.

When exploring different combinations of parameters, combinations can be selected either randomly by random-search or by systematically trying all possible combinations through a grid-search. For faster optimization, a Bayesian Optimization technique has been applied [75]. Bayesian Optimization leverages data from previous training sessions to predict more effective hyperparameter combinations utilizing Bayes Theorem. This project employs Bayesian optimization as the primary method for hyperparameter tuning, aiming to maximize the probability P(metric|params).

The hyperparameter sweep will be conducted using Fold 1 as the test dataset, while Folds 2, 3, 4, and 5 will serve as the training data.

4.5.2 Evaluation

A parameter sweep has been conducted, resulting in a variety of combinations displayed on Figure 4.14 with a best accuracy of 77.73 %. The duration of the sweep with corresponding timestamps and accuracies is depicted in Figure C.1, which can be found in Appendix C.

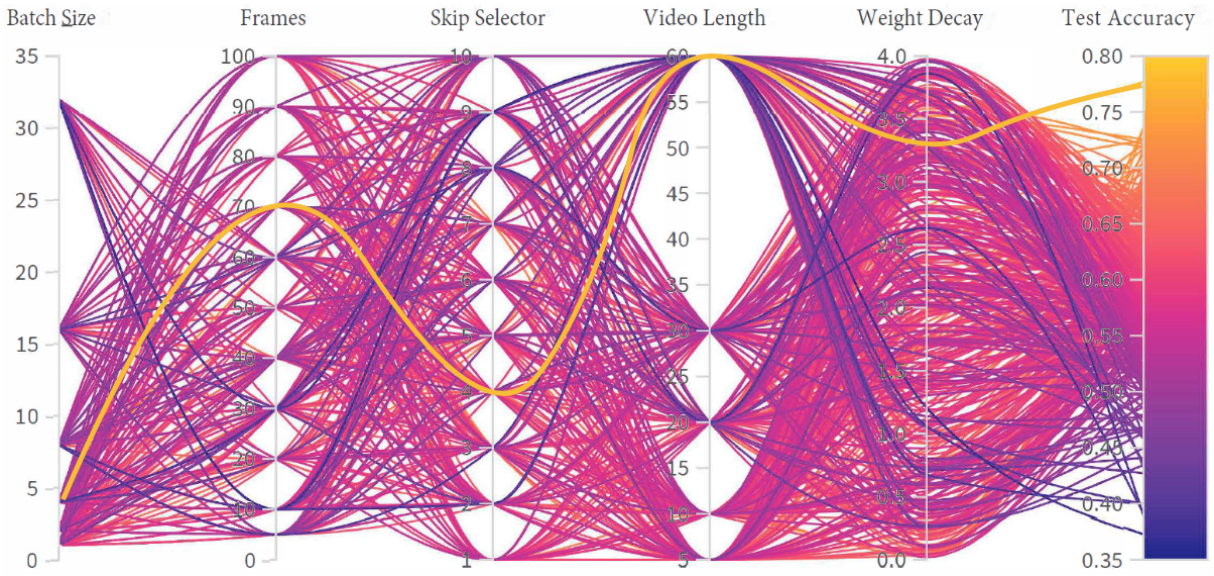


Figure 4.14. The figure displays the a hyperparameter sweep, showcasing 446 unique combinations of variables. Highlighted run displays best test accuracy.

It's observed that there is a pattern for the best performers, indicating which parameters are important. Generally, utilizing more frames seems to indicate better performance. Figure 4.15 portrays the Weights & Biases calculated Importance and Correlation metrics for all 448 runs, indicating which parameters are most important for the sweep. The importance is calculated by training a random forest on the hyperparameter combinations with the testing results as the target output [76]. A high correlation indicates that an increase in a specific hyperparameter value is directly associated with improved test accuracy. Conversely, a low correlation suggests that a decrease in this hyperparameter value is related to improved test accuracy.

Figure 4.15. Weights & Biases importance and correlation metrics for all 448 combinations. A green correlation represents 0 - 1 whereas a red correlation represents -1 - 0.

The Skip Selector has the greatest importance due to the unequal distribution of samples across different video length datasets; for instance, all the 5-second clips are derived from the 60-second clips, among other variations. The correlation is not very high, indicating that it can be beneficial in some cases to skip a large amount of samples - also improving training times. Nonetheless, one could contend that the skip selector might be less significant if its range was to be made narrower. Using only every 10th sample, for instance, could be considered overly aggressive. In contrast, the correlation for the number of frames used is high, indicating that the test accuracy is dependent on a high number of frames sampled by the model.

Yet, as the video size can vary, utilizing 100 frames for a 5 second video may not necessarily be the best case either. Table 4.5 portrays the 12 best performing combinations above 70 % test accuracy, highlighting that the combination with a video length of 60 seconds and sampling 70 frames from the given videos performs best, corresponding to sampling the video with a detail level of 1.167 fps. This is an indicator that the model is performing well sampling less than the 10 fps full detail levels of the videos and sampling $\approx$ 1 fps is considered enough detail to detect drowsiness, indicating that the fine details such as fast blinks are not as important for this particular model.

Batch Size	Frames	Skip Selector	Video Length	Weight Decay	Test Accuracy	fps
4	70	4	60	3.44	0.773	1.167
2	50	3	60	0.8	0.736	0.83
8	70	2	10	0.06	0.736	7
4	60	8	20	4	0.725	3
4	50	2	5	2.79	0.724	10
32	40	8	20	2.103	0.721	2
16	10	4	20	3.614	0.721	0.5
16	90	4	60	3.613	0.718	1.5
4	90	1	60	0.205	0.718	1.5
4	20	4	20	1.714	0.718	1
16	60	1	30	1.439	0.712	2
8	50	1	30	0.226	0.708	1.67

Table 4.5. Summary of Model Parameters and Performance for the 12 best performing and model fps for comparison.

When observing the 12 worst performing in Table 4.6, it's evident that a low fps can have consequences for the performance of the model. However, an important note is that the Batch Size, Skip Selector and Weight Decay all can play an important role, as there are also not well-performing runs with fps ≥ 1. Many of these runs have a high Skip Selector and a Video Length of 60 seconds, indicating a minimal data usage for the given run.

Batch Size	Frames	Skip Selector	Video Length	Weight Decay	Test Accuracy	fps
8	40	7	20	2.306	0.485	2
8	5	6	20	0.4713	0.482	0.25
8	5	9	60	2.202	0.477	0.083
2	40	2	60	1.097	0.477	0.667
32	5	5	60	2.665	0.473	0.083
1	**30**	7	**60**	0.2587	0.45	0.5
4	20	6	60	3.278	0.436	0.333
16	30	2	60	3.707	0.432	0.5
32	**10**	8	**60**	0.7164	0.441	0.167
4	30	8	30	2.631	0.391	1
32	30	9	20	3.863	0.389	1.5
32	10	9	60	1.205	0.373	0.167

Table 4.6. Summary of Model Parameters and Performance for the 12 worst performing and model fps for comparison. Bold marks similar combinations with better performance displayed in Table 4.7.

Furthermore, similar combinations to these runs have shown better performance. An example is the combination in Table 4.7, displaying identical Frame and Video Length combinations, but differs in Batch Size, Skip Selector and Weight Decay.

Batch Size	Frames	Skip Selector	Video Length	Weight Decay	Test Accuracy	fps
16	**10**	4	**60**	0.074	0.6727	0.167
2	**30**	5	**60**	1.368	0.672	0.5

Table 4.7. Model Parameters and Performance for the similar performing model fps for comparison.

Performing k-fold validation on the best performing combination results in an average accuracy of 75.49 %, as displayed in Table 4.8.

Fold	Test Accuracy
1	77.73 %
2	74.29 %
3	76.01 %
4	76.90
5	72.52
Average	**75.49**

Table 4.8. K-fold accuracies and average. Best performing fold is fold 1.

4.6 Concluding remarks

The exploration in this chapter has shown that there is a difference in the amount of detail needed for the TimeSformer model, as more frames and detail put to the model tends to yield better performance. Nevertheless, similar, yet still lower, performance can be attained by supplying the model with a lower overall fps, still underscoring the critical role of temporal relationships in the accurate classification of drowsiness. An average k-fold accuracy of 75.49 % is achieved utilizing two classes, which compared to the original dataset proposal at a testing accuracy of 65.2 % [53] utilizing three classes is not much of an improvement, and lower than the single frame use at 91 % as described in Section 4.2.

Since the TimeSformer model is pretrained on the Google Kinetics 400 dataset — a collection specifically focused on human actions such as shaking hands, dancing salsa, etc. — it is anticipated that pre-training on a broader dataset, particularly one enriched with subtle human movements and facial expressions, or simply training from scratch on a greater drowsiness dataset could significantly enhance its performance. A larger parameter sweep involving more hyperparameters such as amount of trainable parameters could also have been conducted, potentially improving accuracies. Investigating whether a similar pattern would occur on a different dataset type of activity, such as the multi-activity dataset described in Chapter 3, would be interesting as well.

While the TimeSformer model itself may not be practical for direct deployment due to its large size of 121 million parameters, computational demands typical to Transformers and limited computing power in cars, the principles and findings derived from this research can pave the way for more practical implementations through model compression techniques such as model pruning and quantization. Such techniques allow for the model to shrink in size, ultimately reducing the amount of needed computations for it to make a prediction while retaining the capabilities. However, the TimeSformer architecture itself may not be practical as a real world application as it needs a full video for processing, and not just frames fed sequentially in time.

Conclusion 5

This thesis concludes the work done for the last semester of Computer Engineering: AI, Vision and Sound at Aalborg University, with a large part of the work done abroad at LISA and CVRR at UC San Diego. In contrast to a normal semester report approach at Aalborg University, this report is research based, hence the divided independent chapter structure.

The stay at LISA and CVRR in San Diego stay has been rewarding, allowing me to gain insights in a laboratory environment as well as to live abroad. This has been an enriching experience both academically and personally that I find recommendable to try in contrast to the normal pace and style of a semester project at Aalborg University. The stay has resulted in two published papers.

All three chapters contains investigations within the vehicle cabin, and has resulted in three diverse research topics with key findings below:

5.1 Key findings:

Thermal Frame Generation: The study successfully demonstrated that using conditional Generative Adversarial Networks (cGANs), particularly the pix2pix architecture, can generate realistic thermal images from RGB frames. The stacked generation approach proved most effective, highlighting the importance of spatial relationships in the data. Despite these advancements, generalizing the model across different subjects remains a challenge, indicating the need for further research into model customization and fine-tuning for individual drivers.

Driver Activity Classification: The application of Vision-Language models for classifying driver activities showed promising results. By leveraging generalizable representations, the model could classify various in-car activities without extensive tuning. This approach underscores the potential of Vision-Language models for robust driver monitoring systems, capable of adapting to a broad range of tasks and improving safety in autonomous driving environments.

Drowsiness Detection: Utilizing Video Transformers for drowsiness detection demonstrated the feasibility of capturing subtle human movements and facial expressions crucial for this task. Although the TimeSformer model showed potential, it did not perform as well as previous single frame implementations, and its large size and computational demands pose practical challenges for real-world applications.

Bibliography

[1] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[3] J McGinn, Chris Messenger, Michael Williams, and I Heng. Generalised gravitational wave burst generation with generative adversarial networks. *Classical and Quantum Gravity*, 38, 08 2021.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei Efros. Image-to-image translation with conditional adversarial networks. pages 5967–5976, 07 2017.

[5] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. pages 2242–2251, 10 2017.

[6] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications, 2023.

[7] Logitech. Brio-datasheet.pdf. `https://www.logitech.com/content/dam/logitech/vc/en_ch/pdf/Brio-Datasheet.pdf`. (Accessed on 02/03/2024).

[8] Seek Thermal. Micro core specification sheet. `https://www.thermal.com/uploads/1/0/1/3/101388544/micro_core_specification_sheet_v2-2021_1.pdf`. (Accessed on 02/03/2024).

[9] George T. Heineman, Gary Pollice, and Stanley Selkow. *Algorithms in a Nutshell*. O'Reilly Media, Inc., 2009.

[10] Anjan Debnath, Tèmítáyò Olówu, Imtiaz Parvez, and Arif Sarwat. A binary search algorithm based optimal sizing of photovoltaic and energy storage systems. pages 563–568, 04 2021.

[11] Erik Linder-Norén. eriklindernoren/pytorch-gan: Pytorch implementations of generative adversarial networks. `https://github.com/eriklindernoren/PyTorch-GAN`. (Accessed on 11/04/2024).

[12] Erik Linder-Norén. eriklindernoren/keras-gan: Keras implementations of generative adversarial networks. `https://github.com/eriklindernoren/Keras-GAN`. (Accessed on 11/04/2024).

[13] pix2pix: Image-to-image translation with a conditional gan | tensorflow core. `https://www.tensorflow.org/tutorials/generative/pix2pix`. (Accessed on 11/04/2024).

[14] Cyclegan | tensorflow core. `https://www.tensorflow.org/tutorials/generative/cyclegan`. (Accessed on 11/04/2024).

[15] Smoothing | weights & biases documentation. `https://docs.wandb.ai/guides/app/features/panels/line-plot/smoothing#running-average`. (Accessed on 11/04/2024).

[16] Yunpeng Li, Dominik Roblek, and Marco Tagliasacchi. From here to there: Video inbetweening using direct 3d convolutions. *arXiv preprint arXiv:1905.10240*, 2019.

[17] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher J. Pal. Robust motion in-betweening. *CoRR*, abs/2102.04942, 2021.

[18] Kurt M. DeGoede, James A. Ashton-Miller, Jimmy M. Liao, and Neil B. Alexander. How Quickly Can Healthy Adults Move Their Hands to Intercept an Approaching Object? Age and Gender Effects. *The Journals of Gerontology: Series A*, 56(9):M584–M588, 09 2001.

[19] Shane Barratt and Rishi Sharma. A note on the inception score, 2018.

[20] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *CoRR*, abs/1812.01717, 2018.

[21] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications, 2024.

[22] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023.

[23] Wei Zhou, Yinlong Qian, Zequn Jie, and Lin Ma. Multi view action recognition for distracted driver behavior localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 5375–5380, June 2023.

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[25] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[26] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. 2021.

[27] Dan Fu, Mayee Chen, Megan Leszczynski, and Chris Ré. Advances in understanding, improving, and applying contrastive learning · hazy research. https://hazyresearch.stanford.edu/blog/2022-04-19-contrastive-1. (Accessed on 06/05/2024).

[28] Sherman E. Lo. Mode filter - sherman lo. https://warwick.ac.uk/fac/sci/statistics/staff/research_students/ip/postphd/, 2020. (Accessed on 08/05/2024).

[29] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding, 2023.

[30] Fatigue and crash risk - european commission. https://road-safety.transport.ec.europa.eu/european-road-safety-observatory/statistics-and-analysis-archive/fatigue/fatigue-and-crash-risk_en. (Accessed on 16/02/2024).

[31] Drowsy driving: Avoid falling asleep behind the wheel | nhtsa. https://www.nhtsa.gov/risky-driving/drowsy-driving. (Accessed on 16/02/2024).

[32] Toshiya Arakawa. Trends and future prospects of the drowsiness detection and estimation technology. *Sensors*, 21(23), 2021.

[33] Driving time and rest periods - european commission. https://transport.ec.europa.eu/transport-modes/road/social-provisions/driving-time-and-rest-periods_en. (Accessed on 15/04/2024).

[34] Interstate truck driver's guide to hours of service. https://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/Drivers%20Guide%20to%20HOS%202015_508.pdf. (Accessed on 15/04/2024).

[35] J.D. Levine. A road to injustice paved with good intentions: Maggie's misguided crackdown on drowsy driving. *The Hastings law journal*, 56:1297–1315, 06 2005.

[36] Explanatory memorandum. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=PI_COM:Ares(2021)1075107&rid=11. (Accessed on 22/04/2024).

[37] European Commission. Eu road safety policy framework 2021-2030 – next steps towards "vision zero", 2019. (Accessed on 22/04/2024).

[38] A M Williamson and Anne-Marie Feyer. Moderate sleep deprivation produces impairments in cognitive and motor performance equivalent to legally prescribed levels of alcohol intoxication. *Occupational and Environmental Medicine*, 57(10):649–655, 2000.

[39] Impaired driving laws, enforcement and prevention | us department of transportation. https://www.transportation.gov/mission/health/Impaired-Driving-Laws-Enforcement-and-Prevention. (Accessed on 15/04/2024).

[40] Peter Anderson, Lars Møller, and Gauden Galea. Alcohol in the european union: consumption, harm and policy approaches, 2012.

[41] R Hingson. Prevention of drinking and driving. *Alcohol Health Res World*, 20(4):219–226, 1996.

[42] Yaman Albadawi, Maen Takruri, and Mohammed Awad. A review of recent developments in driver drowsiness detection systems. *Sensors*, 22(5), 2022.

[43] Driver drowsiness detection. https://www.bosch-mobility.com/en/solutions/assistance-systems/driver-drowsiness-detection/. (Accessed on 23/02/2024).

[44] Furkat Safarov, Farkhod Akhmedov, Akmalbek Bobomirzaevich Abdusalomov, Rashid Nasimov, and Young Im Cho. Real-time deep learning-based drowsiness detection: Leveraging computer-vision and eye-blink analyses for enhanced road safety. *Sensors*, 23(14), 2023.

[45] Robin Dua, Andrei Broder, and Vidhya Navalpakkam. Operator drowsiness test. https://www.tdcommons.org/dpubs_series/1854, January 2019.

[46] S. Nordbakke and F. Sagberg. Sleepy at the wheel: Knowledge, symptoms and behaviour among car drivers. *Transportation Research Part F: Traffic Psychology and Behaviour*, 10(1):1–10, 2007.

[47] Reza Ghoddoosian, Marnim Galib, and Vassilis Athitsos. A realistic dataset and baseline temporal model for early drowsiness detection. *CoRR*, abs/1904.07312, 2019.

[48] A Miley, G Kecklund, and T Åkerstedt. Comparing two versions of the karolinska sleepiness scale (kss). *Sleep Biol Rhythms*, 14(3):257–260, 2016.

[49] How does mercedes benz attention assist work? | technology overview. https://www.mercedesbenzofeaston.com/mercedes-benz-attention-assist/%7D%7D. (Accessed on 22/04/2024).

[50] Driver alert system | volkswagen newsroom. https://www.volkswagen-newsroom.com/en/driver-alert-system-3932. (Accessed on 22/04/2024).

[51] Bmw model upgrade measures taking effect from the summer of 2013. https://www.press.bmwgroup.com/global/article/detail/T0141144EN/bmw-model-upgrade-measures-taking-effect-from-the-summer-of-2013. (Accessed on 22/04/2024).

[52] Ford's wake-up call for europe's sleepy drivers. https://web.archive.org/web/20110513232258/http://media.ford.com/article_print.cfm?article_id=34562. (Accessed on 22/04/2024).

[53] Reza Ghoddoosian, Marnim Galib, and Vassilis Athitsos. A realistic dataset and baseline temporal model for early drowsiness detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[54] Shabnam Abtahi, Mona Omidyeganeh, Shervin Shirmohammadi, and Behnoosh Hariri. Yawdd: a yawning detection dataset. In *Proceedings of the 5th ACM Multimedia Systems Conference*, MMSys '14, page 24–28, New York, NY, USA, 2014. Association for Computing Machinery.

[55] Pengkun Liu, Hung-Lin Chi, Xiao Li, and Jingjing Guo. Effects of dataset characteristics on the performance of fatigue detection for crane operators using hybrid deep neural networks. *Automation in Construction*, 132:103901, 2021.

[56] Reza Tamanani, Radu Muresan, and Arafat Al-Dweik. Estimation of driver vigilance status using real-time facial expression and deep learning. *IEEE Sensors Letters*, 5(5):1–4, 2021.

[57] Ghanta Sai Krishna, Kundrapu Supriya, Jai Vardhan, and Mallikharjuna Rao K. Vision transformers and yolov5 based driver drowsiness detection framework, 2022.

[58] Ismail Nasri, Mohammed Karrouchi, Hajar Snoussi, Kamal Kassmi, and Abdelhafid Messaoudi. Detection and prediction of driver drowsiness for the prevention of road accidents using deep neural networks techniques. In Saad Bennani, Younes Lakhrissi, Ghizlane Khaissidi, Anass Mansouri, and Youness Khamlichi, editors, *WITS 2020*, pages 57–64, Singapore, 2022. Springer Singapore.

[59] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I, 2001.

[60] Rafael Padilla, Cicero Filho, and Marly Costa. Evaluation of haar cascade classifiers for face detection. 04 2012.

[61] Opencv: Face detection using haar cascades. `https://docs.opencv.org/4.x/d2/d99/tutorial_js_face_detection.html`. (Accessed on 27/02/2024).

[62] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019.

[63] Sefik Ilkin Serengil and Alper Ozpinar. Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4. IEEE, 2021.

[64] Jaehong Yu, Seoung Bum Kim, Jinli Bai, and Sung Won Han. Comparative study on exponentially weighted moving average approaches for the self-starting forecasting. *Applied Sciences*, 10(20), 2020.

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[66] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[67] OpenAI. Gpt-4 technical report. *ar5iv*, 2023. Accessed: 2024-03-26.

[68] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. *CoRR*, abs/1711.11248, 2017.

[69] Atiq Rehman and Samir Brahim Belhaouari. Deep learning for video classification: A review. 08 2021.

[70] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *CoRR*, abs/2102.05095, 2021.

[71] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[72] Wen Wang, Jing Zhang, Yang Cao, Yongliang Shen, and Dacheng Tao. Towards data-efficient detection transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 88–105, Cham, 2022. Springer Nature Switzerland.

[73] Weights & Biases. Home – weights & biases. `https://wandb.ai/home`. (Accessed on 01/03/2024).

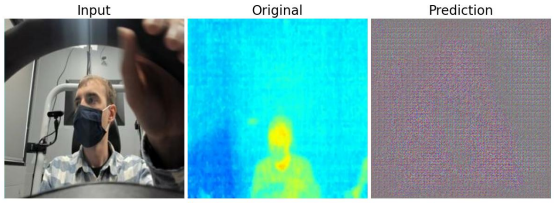
[74] Jaejun Lee, Raphael Tang, and Jimmy Lin. What would elsa do? freezing layers during transformer fine-tuning. *CoRR*, abs/1911.03090, 2019.

[75] Sayak Paul. Bayesian hyperparameter optimization - a primer on weights & biases. `https://wandb.ai/site/articles/bayesian-hyperparameter-optimization-a-primer`. (Accessed on 01/03/2024).

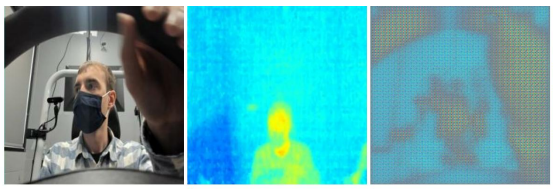
[76] Parameter importance | weights & biases documentation. `https://docs.wandb.ai/guides/app/features/panels/parameter-importance`. (Accessed on 02/05/2024).

[77] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

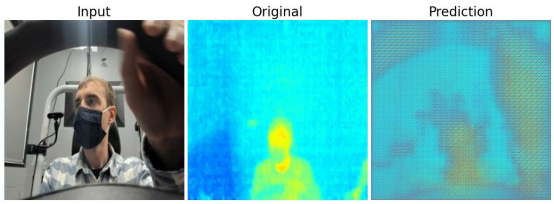
Thermal From RGB Genration Results Extra Material A



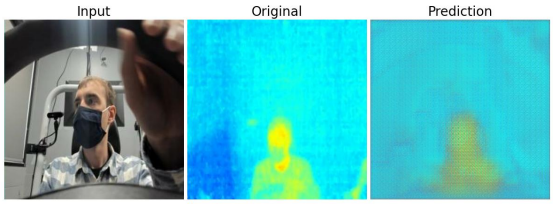
Step: 0, L1 Loss: 0.886.



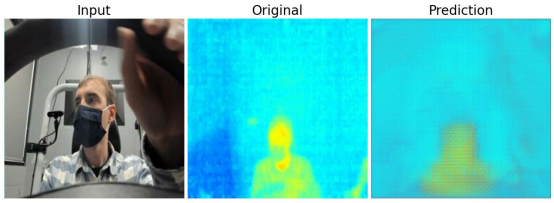
Step: 10, L1 Loss: 0.683.



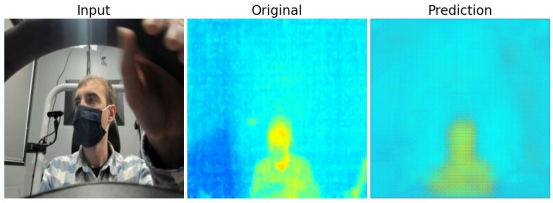
Step: 20, L1 Loss: 0.487.



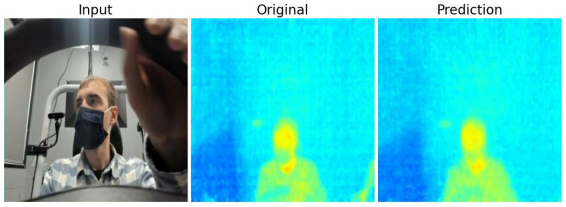
Step: 30, L1 Loss: 0.346.



Step: 40, L1 Loss: 0.239.



Step: 50, L1 Loss: 0.210.



Step: 20000, L1 Loss: 0.103.

Figure A.1. From top to bottom, the generator output at different iterations of training are shown. Originally, the generator produces a random image, and refines its output to match the intended thermal image over time. These images are separated by only 10 iterations each (from 0 to 50), except for the final image which represents a jump to 20,000 iterations.

UTA-RLDD Analysis Extra Material B

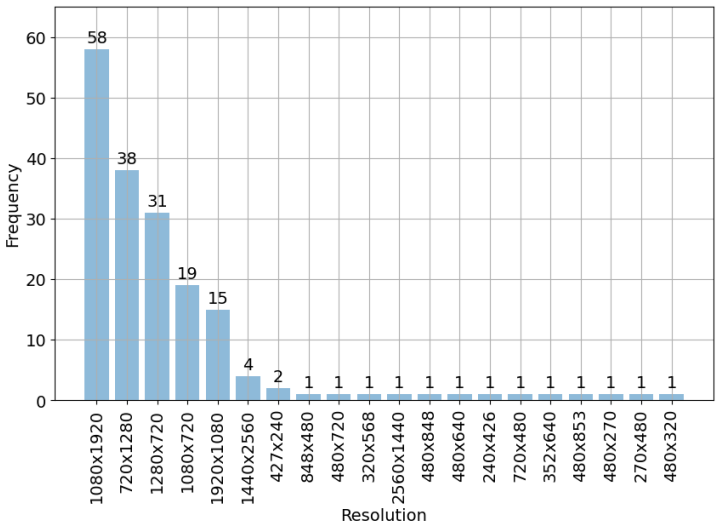


Figure B.1. Histogram illustrating the distribution frame resolution in the dataset.

Class	Duration (h)
Alert	10.40
Low Vigilance	10.49
Drowsy	10.28
Total duration	**31.17**

Table B.1. Total duration of each class in hours (h) for UTA-RLDD.

Drowsiness Extra Material C

Component	Subcomponent	Specifications
Embeddings	- Patch Embeddings	Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
	- Positional Dropout	Dropout(p=0.0)
	- Temporal Dropout	Dropout(p=0.0)
Encoder Layer (0-11)	- Self-Attention	
	- Query/Key/Value	Linear(768, 2304)
	- Attention Dropout	Dropout(p=0.0)
	- Projection	Linear(768, 768)
	- Intermediate	
	- Dense Layer	Linear(768, 3072)
	- Activation	GELU
	- Dropout	Dropout(p=0.0)
	- Output	
	- Dense Layer	Linear(3072, 768)
	- Dropout	Dropout(p=0.0)
	- Layer Normalization	
	- Before Attention	LayerNorm(768)
	- After Attention	LayerNorm(768)
	- Temporal LayerNorm	LayerNorm(768)
	- Temporal Attention	
	- Self-Attention	
	- Query/Key/Value	Linear(768, 2304)
	- Attention Dropout	Dropout(p=0.0)
	- Projection	Linear(768, 768)
	- Dense Layer	Linear(768, 768)
	- Dropout	Dropout(p=0.0)
Final Layer Normalization		LayerNorm(768)
Classifier	- Linear Layer	Linear(768, 2)

Table C.1. The utilized TimeSformer model employs a series of components designed to process video data through spatial and temporal features extraction. The *Embeddings* section projects video frames into a high-dimensional space and applies dropout to reduce overfitting. Each *Encoder Layer* repeats 12 times, where self-attention mechanisms compute interactions within frames, and temporal attention aggregates these across time. GELU activations introduce non-linearity between dense layers that expand and contract feature representations [77]. Layer normalization is applied throughout to stabilize learning. The *Classifier* at the end predicts classes based on the extracted features.

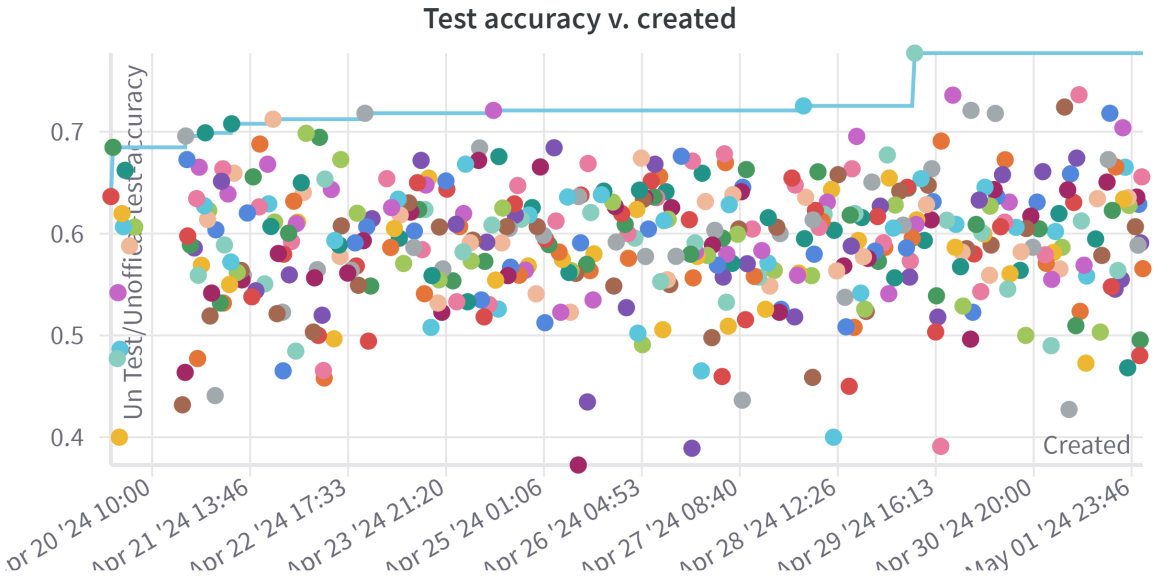


Figure C.1. Test accuracy v. created. 446 runs.

Submitted papers D

The following pages contain the scientific articles that were submitted related to this project and a co-author permission letter from Ross Greer. The papers have been accepted, although they may be subject to further revisions following the submission of this project.

Present items in this Appendix:

- Learning to Find Missing Video Frames with Synthetic Data Augmentation: A General Framework and Application in Generating Thermal Images Using RGB Cameras. Accepted at the 35th IEEE Intelligent Vehicles Symposium (IV).
- Driver Activity Classification Using Generalizable Representations from Vision-Language Models. Accepted at the Computer Vision and Pattern Recognizion (CVPR) Vision and Language for Autonomous Driving and Robotics Workshop.
- Co-Author Permission Letter from Ross Greer.

Learning to Find Missing Video Frames with Synthetic Data Augmentation: A General Framework and Application in Generating Thermal Images Using RGB Cameras

Mathias Viborg Andersen
Laboratory for Intelligent & Safe Automobiles (LISA)
University of California San Diego
mvan19@student.aau.dk

Ross Greer
Laboratory for Intelligent & Safe Automobiles (LISA)
University of California San Diego
regreer@ucsd.edu

Andreas Møgelmose
Visual Analysis and Perception Lab
Aalborg Universitet
anmo@create.aau.dk

Mohan M. Trivedi
Laboratory for Intelligent & Safe Automobiles (LISA)
University of California San Diego
mtrivedi@ucsd.edu

***Abstract*—Advanced Driver Assistance Systems (ADAS) in intelligent vehicles rely on accurate driver perception within the vehicle cabin, often leveraging a combination of sensing modalities. However, these modalities operate at varying rates, posing challenges for real-time, comprehensive driver state monitoring. This paper addresses the issue of missing data due to sensor frame rate mismatches, introducing a generative model approach to create synthetic yet realistic thermal imagery. We propose using conditional generative adversarial networks (cGANs), specifically comparing the pix2pix and CycleGAN architectures. Experimental results demonstrate that pix2pix outperforms CycleGAN, and utilizing multi-view input styles, especially stacked views, enhances the accuracy of thermal image generation. Moreover, the study evaluates the model's generalizability across different subjects, revealing the importance of individualized training for optimal performance. The findings suggest the potential of generative models in addressing missing frames, advancing driver state monitoring for intelligent vehicles, and underscoring the need for continued research in model generalization and customization.**

***Index Terms*—image synthesis, generative artificial intelligence, thermal imagery, pseudo-labeled dataset, data augmentation**

I. Introduction

Many advanced driver assistance systems (ADAS) rely on accurate perception of the human driver by looking inside the intelligent vehicle cabin [1], [2]. No single-view or sensing modality perfectly suits every task or design specification, so often a combination of views can be leveraged for enhanced driver state understanding [3]. However, different sensing modalities operate at different rates and often perform similarly to RGB using neural networks [4], [5]. For example, thermal cameras often operate at a fraction of the rate of cameras on the visible light spectrum, but are useful toward a variety of tasks in driver state monitoring by providing useful information for understanding occlusion, circulation, respiration, and other heat-related observations [6]–[11].

In many cases, it is useful for models to make observations based on a sequence of observations, rather than a single instance in time; doing so allows the modeling of dynamic events and driver actions [12]–[14], as well as inference reinforcement by repeated agreement between information observed at nearby times [15]. Considering again the multimodal nature of driver observation systems, misaligned sensor rates are not a problem when data from each modality is considered individually, but requires careful synchronization and creative approaches to modeling in cases when data from one sensor may not be provided at the ideal rate for best utilizing the remaining sensors. We illustrate this problem, and associated problems and trade-offs, in Figure 1; essentially, one can either use a high-frequency model with intermittent missing data from low-frequency sensors, or a low-frequency model which may sacrifice temporal precision of inference. The risk of using low-frequency sensors is the mismatch that occurs when the driver takes an action which changes state in a way that can be observed by one sensor but missed by another; this can create conflicting information and multimodal model confusion if not handled properly, illustrated in Figure 2.

In this research, we propose a solution using a generative model to create pseudo-complete data samples, pairing both real (low frequency) thermal imagery and generated samples

M. V. Andersen, R. Greer, and M. Trivedi are with the Laboratory for Intelligent and Safe Automobiles at University of California San Diego. A. Møgelmose is with the Visual Analysis and Perception Lab at Aalborg University.

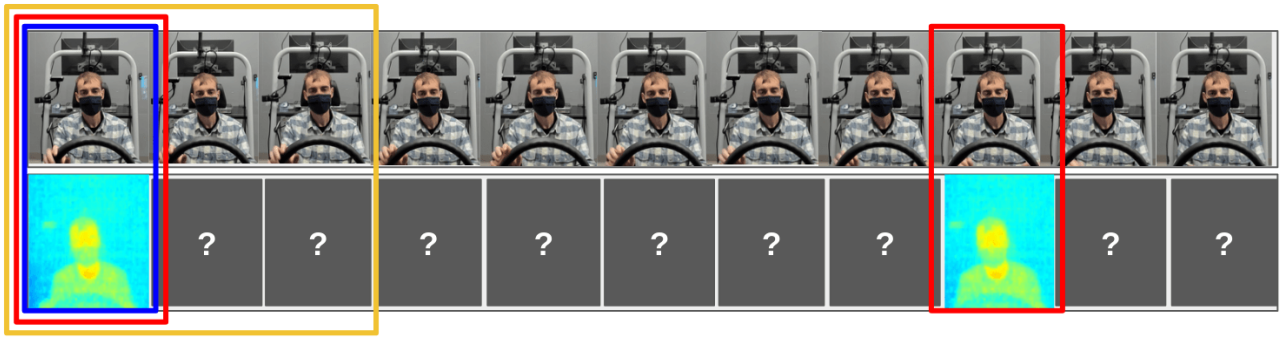


Fig. 1: Many perspectives and modalities of data may contribute to robust driver state monitoring. Differing frame rates of sensors lead to an unavailability of "complete" sets of data from all modalities for a given instance. Because many driver states are best inferred from temporal patterns, an ideal data stream would have constant availability of all sources at each instance. Without such a stream, models may be limited to instance inference (blue), complete-but-temporally-distant sequences (red), or incomplete-but-temporally-local sequences (yellow). By generating missing data, we can provide synthetic but useful representations to fill in these gray gaps, enabling accurate downstream state estimation models using pseudo-complete, temporally-local sequences.

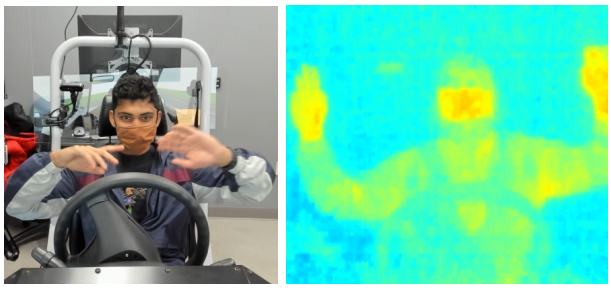


Fig. 2: When sensors operate at different rates, it is possible that the temporally-nearest measurement to a given instance may have taken place before a significant action for one sensor, and after the action for another. In the above example, the driver has abruptly moved his hands closer to the wheel; however, the thermal camera has not yet processed another signal to capture this motion. So, if both "most recent" signals are sent to a multimodal model meant to estimate a driver's takeover readiness (e.g. proximity of hands to the steering wheel), the model would have a large amount of uncertainty from modal disagreement.

with high-frequency visible light images captured from multiple perspectives.

II. Related Research

A. Synthetic Thermal Images for Data Augmentation

Various modes of image data, from RAW [4], [5] to RGB to IR, and even to non-light-based imagery such as temperature, provide utility in a variety of applications; further, cross-modality image synthesis is useful as a data augmentation strategy in a variety of tasks, such as terrain classification (visible to IR) [16], tissue segmentation (MRI to CT) [17], and heart observation (various CMR medical imaging techniques) [18]. Specific to human interaction, Hermosilla et al. [19] generate thermal facial images from noise using StyleGAN2, in order to build a robust dataset with tunable features that can be used to train deep learning models to detect faces within thermal images, a task which is generally more difficult than detection from visible light images but may be useful in certain sensing environments. Our research similarly utilizes GAN as the underlying generative learning paradigm, with the common goal of augmenting datasets towards problems in human subject feature extraction and understanding.

B. Translating between Thermal and Visible Light

Deep learning has revolutionized image processing and opened up new possibilities for image generation. Deep learning models, particularly generative adversarial networks (GANs), have demonstrated remarkable capabilities in translating images between different domains. Relevant to our domains of interest (visible light and thermal signature), Abdrakhmanova et al. [20] create the SpeakingFaces dataset and explore models to generate visible images from thermal images, since facial features for landmark recognition and detection are too obscure in thermal imagery, reducing possible use cases in human-computer-interaction (HCI), biometric authentication, and other systems. They use CycleGAN and CUT models to map thermal face images to the visual spectrum, evaluating both the generated Fréchet inception distance as well as the ability of the generated models to produce the correct output on downstream facial landmark detection tasks.

Li et al. [21] introduced the dual-attention generative adversarial network (DAGAN) for the generation of thermal images from visible light, but outside the realm of human subjects and in the domain of fire safety, useful in estimating locations and temperatures of flames in room fires. Though the temperature range, contrast, and locality on human subjects is significantly different from that of an open flame, their research provides

a strong indication that GAN architectures are suited to the task.

C. Predicting Missing Frames

For real-world data, the expected translational motion of objects between frames has allowed for missing individual frames to be estimated using a Kalman filtering approach [22], and for larger segments of up to 14 frames to be created from a fully convolutional model [23]. However, such methods which effectively "in-paint" or "in-between" video sequences are ineffective toward replacing larger missing sequences (in this case, approximately 1 frame available in every 5-frame period). Further exacerbating this issue, the frame rate is slow enough that basic human motion changes state at a scale faster than the sensor records– meaning that even knowing the surrounding frames may be insufficient to fill in the activity of the frames in between. Fortunately, in this problem setting, we have at our disposal additional information from the "missing" times, rather than just its surrounding pieces.

III. Methods

A. Synthetic Data Augmentation

In the case presented in this paper, we consider a downstream goal task of driver state estimation, and we have a set of thermal and visible light images which are used to supervise the training of this task. However, we also have a set of visible light images with no matching thermal imagery; by training an intermediate model which generates associated thermal images, and then using these generated images in the training of a driver state estimation model, we can augment the dataset to enable the use of models which operate over complete sets of sensor data at a high frequency.

B. Algorithm

We train and evaluate two conditional generative adversarial network (cGAN) architectures [24]; the pix2pix architecture [25] and the CycleGAN architecture [26]. These models consist of two competing neural networks: a generator that produces thermal images from RGB inputs and a discriminator that distinguishes between real thermal images and generated ones. Through adversarial training, the generator learns to synthesize realistic thermal images that are indistinguishable from real ones. The cGAN architecture of pix2pix is seen in Figure 3, and a demonstration of its iterative learning is shown in Figure 6.

Our output for all methods is a thermal image, like the example shown in Figure 5a. We evaluate three types of input for experimental comparison:

1) Front-View RGB (Figure 5b)
2) Single-Subject Four-View RGB, tesselated (Figure 5c)
3) Single-Subject Four-View RGB, stacked (Figure 5d)

Though these input styles may appear unusual, their compactness allows for training on a single GPU system, and on an efficient model architecture, as opposed to creating three additional convolutional "heads" to extract features from the individual images. Though there are some false edges introduced due to the stacking structures, our results show that these collage images are still able to significantly outperform single-view learning. Additionally, we create one further experiment to evaluate for the effects of single-subject training versus multi-subject training for the Front View.

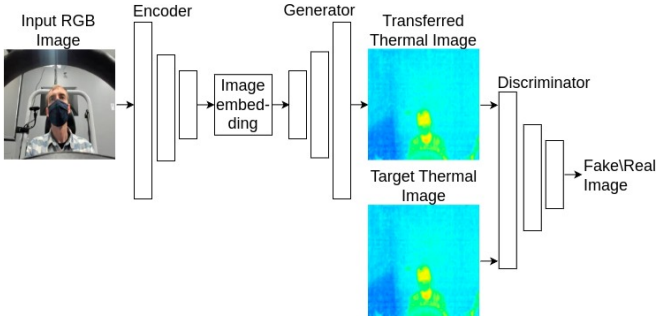


Fig. 3: The flow of pix2pix applied in this work.

C. Dataset

The dataset utilized in this study is notably comprehensive, incorporating various perspectives beneficial for developing and evaluating innovative approaches to generating thermal images from RGB inputs as well as driver state monitoring in general. It consists of distinct viewpoints, including *thermal*, *front*, *overhead*, *profile*, and *tablet* orientations, as displayed in Figure 4.

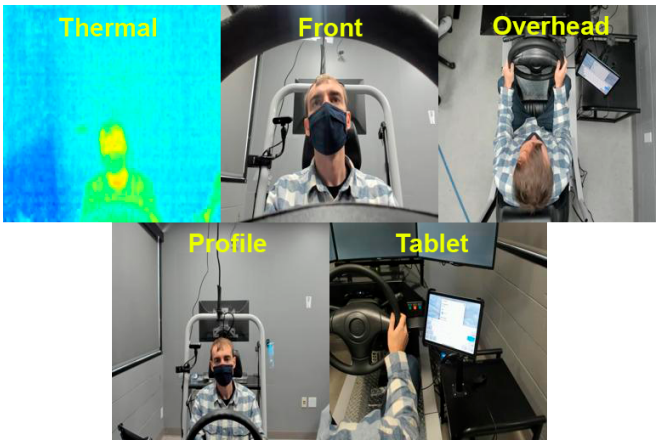
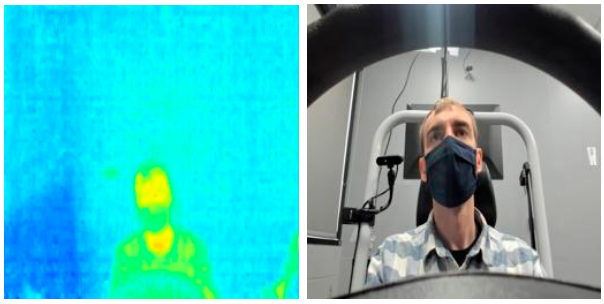
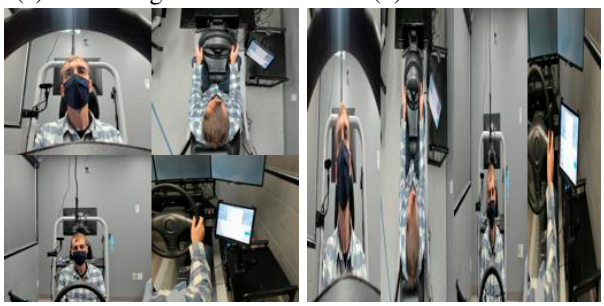


Fig. 4: Example images showcasing perspectives captured from used cameras within our simulator setup.

The dataset consists of captures of 17 subjects seated at a simulated driver's seat. Notably, the RGB images are captured at a rate of approximately 30 frames per second (fps), while the thermal imaging data is sourced from a thermal camera operating at less than 9 fps, and represent a range of -20°C to 300°C, scaled to [0, 255]. Thorough synchronization and preparation procedures have been applied to ensure the optimal integration of the multi-view data.



(a) Thermal ground truth. (b) Front-View.



(c) Four-View, Tessellated. (d) Four-View, Stacked.

Fig. 5: In our experiments, different inputs are evaluated on their potential for generating an image similar to the thermal ground truth.

TABLE I: Comparison of Model Architectures when training front-view.

Method	Average Test L1 Error	Standard Deviation
CycleGAN	0.1644	0.0585
pix2pix	**0.0676**	**0.0106**

From each subject we create a collection of 500 thermal + RGB image synchronous groups. In each experiment, an allocation of 80 % for training, 10 % for validation, and 10 % for testing has been used. In the case of training on the aggregate pool of all subjects, we randomly select 5,000 of the 8,500 samples to account for our available training hardware.

IV. Experimental Evaluation

A. Generative Architecture

We first compare the performance of the pix2pix architecture versus the CycleGAN architecture, which has proven useful in past data augmentation for driver state monitoring [27]. As shown in Table I, the performance of the CycleGAN is significantly subpar to pix2pix. We expect that this is due to the attempt by CycleGAN to reconstruct the original input from its generated output during training; because the thermal image is significantly lossy compared to the amount of information in the visible light images, this relationship is more difficult to model beyond approximation. For interpretation, we note that all experimental error values are on normalized pixel values in the range [0, 1], meaning that most errors range around 5-6% of the pixel range.

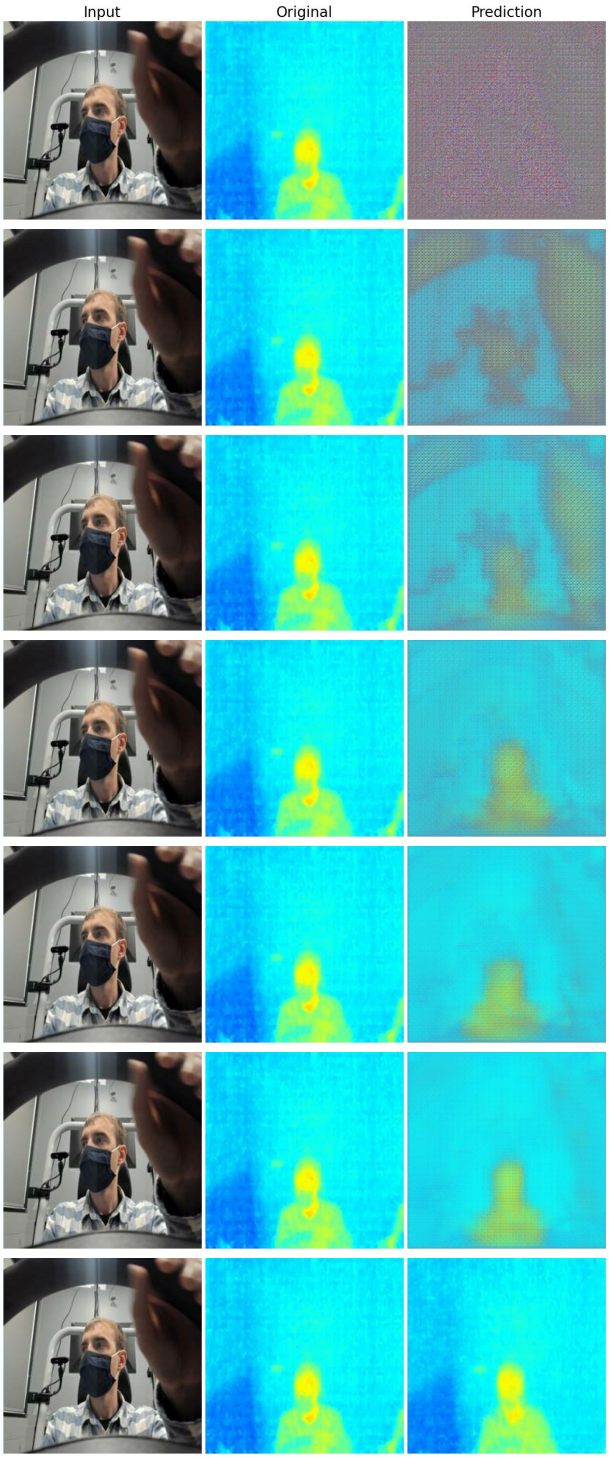


Fig. 6: From top to bottom, we show the generator output at different iterations of training. Originally, the generator produces a random image, and refines its output to match the intended thermal image over time. These images are separated by only 10 iterations each, except for the final image which represents a jump to 20,000 iterations.

TABLE II: Comparison of Input Style; Average Performance Across 17 Subjects.

Dataset	Average Test L1 Error	Standard Deviation
Front-View	0.0676	0.0106
Four-View, Tessellated	0.0587	0.0109
Four-View, Stacked	**0.0559**	**0.0093**

TABLE III: Comparison of Single vs. Multi-Subject Training on Front View.

Dataset	Average Test L1 Error	Std. Deviation
Single-Subject Training	**0.0676**	**0.0106**
Multi-Subject Training	0.1116	0.0186

B. Input Style

The results of our experiments comparing the three different styles of input are presented in Table II. We find that a combination of views outperforms generation from a single-view, Figure 9, (perhaps assisting in understanding hand and posture positioning and respective heat signatures), and that the stacked-view and tesselated-view, Figures 7&10, of the images provides an efficient and effective input, evidenced by the lowest average L1 error (0.0559) and a comparatively low standard deviation (0.0093). This suggests that considering spatial relationships by multi-view information enhances the model's accuracy in thermal image generation, as seen in Figure 7.

C. Subject Generalizability

The results of our experiments on model generalizability to multi-subject training data is presented in Table III. We find that though the training dataset size grows significantly ($17\times$) and still includes the original training data, the additional subjects seem to contribute more to model confusion than generalized pattern creation, showing a higher average L1 error of 0.1116 and supporting the idea that these models are best trained on an individual basis. The relatively weak performance when trained on the more diverse set of data can be observed in Figure 8.

V. Concluding Remarks

Our study on generating thermal images from RGB counterparts highlights promising results with effective prediction approximation. However, the persisting challenge of the missing frames issue underscores the complexity of the task, demanding further research, as predictions must be near perfect. The stacked generation approach proved most successful, emphasizing the importance of spatial relationships. Despite these advancements, the model's generalization between subjects remains the worst performer, warranting continued efforts for improved adaptability across diverse scenarios so that singular models may be trained and deployed, and motivating research into the potential of small-data fine-tuning for model customization to individual drivers. Another venue of continued research is to analyze this approach in different tasks such as utilizing Lidar data, different environments from inside a vehicle cabin and verify utilizing more datasets. This, aswell as taking account for potential missing frames within the four utilized RGB views [28], [29]. The value of observing time-varying patterns is beneficial to many autonomous driving applications [30]–[32], providing a means to infer useful, high-frequency cues from temporal dynamics of the observed subject.

In summary, our generative approach shows potential to address the missing frames problem (caused by sensor frame rate mismatches and intermittent failures), providing a means for higher frequency driver state monitoring for enhanced intelligent vehicle awareness and rapid, safe decision-making.

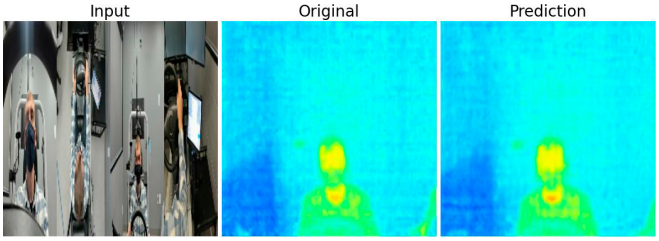


Fig. 7: Single-Subject Four-View, Stacked Prediction Example.

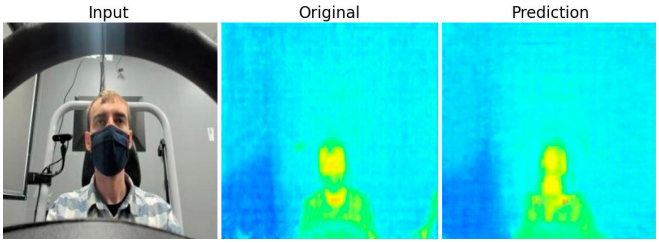


Fig. 8: Multi-Subject Front-View Prediction Example.

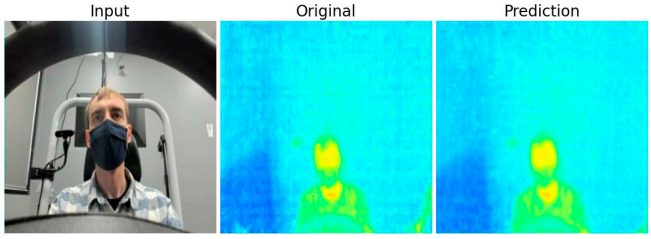


Fig. 9: Single-Subject Front-View Prediction Example.

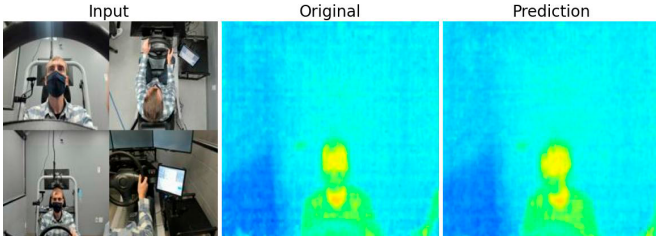


Fig. 10: Four-View, Tessellated Prediction Example.

ACKNOWLEDGEMENTS

The authors would like to thank Sumega Mandadi for her assistance in the synchronization and management of the experimental dataset.

REFERENCES

[1] S. Vora, A. Rangesh, and M. M. Trivedi, "Driver gaze zone estimation using convolutional neural networks: A general framework and ablative analysis," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 254–265, 2018.

[2] E. Ohn-Bar, A. Tawari, S. Martin, and M. M. Trivedi, "On surveillance for safety critical events: In-vehicle video networks for predictive driver assistance systems," *Computer Vision and Image Understanding*, vol. 134, pp. 130–140, 2015.

[3] R. Greer, L. Rakla, A. Gopalkrishnan, and M. Trivedi, "Multi-view ensemble learning with missing data: Computational framework and evaluations using novel data from the safe autonomous driving domain," *arXiv preprint arXiv:2301.12592*, 2023.

[4] C. Kantas, B. Antoniussen, M. V. Andersen, R. Munksø, S. Kotnala, S. B. Jensen, A. Møgelmose, L. Nørgaard, and T. B. Moeslund, "Raw instinct: Trust your classifiers and skip the conversion," in *2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, pp. 456–460, IEEE, 2023.

[5] R. Munksø, M. Andersen, L. Nørgaard, A. Møgelmose, and T. Moeslund, "Enabling raw image classification using existing rgb classifiers," pp. 123–129, 01 2024.

[6] S. Kajiwara, "Driver-condition detection using a thermal imaging camera and neural networks," *International journal of automotive technology*, vol. 22, pp. 1505–1515, 2021.

[7] S. Bole, C. Fournier, C. Lavergne, G. Druart, and T. Lépine, "Driver head pose tracking with thermal camera," in *Infrared Sensors, Devices, and Applications VI*, vol. 9974, pp. 158–167, SPIE, 2016.

[8] V. Mattioli, L. Davoli, L. Belli, G. Ferrari, and R. Raheli, "Thermal camera-based driver monitoring in the automotive scenario," in *2023 AEIT International Conference on Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE)*, pp. 1–6, IEEE, 2023.

[9] S. E. H. Kiashari, A. Nahvi, A. Homayounfard, and H. Bakhoda, "Monitoring the variation in driver respiration rate from wakefulness to drowsiness: a non-intrusive method for drowsiness detection using thermal imaging," *Journal of Sleep Sciences*, vol. 3, no. 1-2, pp. 1–9, 2018.

[10] C. Weiss, A. Kirmas, S. Lemcke, S. Böshagen, M. Walter, L. Eckstein, and S. Leonhardt, "Head tracking in automotive environments for driver monitoring using a low resolution thermal camera," *Vehicles*, vol. 4, no. 1, pp. 219–233, 2022.

[11] C. Palmero, A. Clapés, C. Bahnsen, A. Møgelmose, T. B. Moeslund, and S. Escalera, "Multi-modal rgb–depth–thermal human body segmentation," *International Journal of Computer Vision*, vol. 118, pp. 217–239, 2016.

[12] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 231–236, IEEE, 2021.

[13] A. Rangesh, N. Deo, R. Greer, P. Gunaratne, and M. M. Trivedi, "Predicting take-over time for autonomous driving with real-world data: Robust data augmentation, models, and evaluation," *arXiv preprint arXiv:2107.12932*, 2021.

[14] R. Greer, N. Deo, A. Rangesh, P. Gunaratne, and M. Trivedi, "Safe control transitions: Machine vision based observable readiness index and data-driven takeover time prediction," *arXiv preprint arXiv:2301.05805*, 2023.

[15] R. Greer, L. Rakla, A. Gopalan, and M. Trivedi, "(safe) smart hands: Hand activity analysis and distraction alerts using a multi-camera framework," *arXiv preprint arXiv:2301.05838*, 2023.

[16] Y. Iwashita, K. Nakashima, S. Rafol, A. Stoica, and R. Kurazume, "Mu-net: Deep learning-based thermal ir image estimation from rgb image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[17] X. Chen, C. Lian, L. Wang, H. Deng, T. Kuang, S. H. Fung, J. Gateno, D. Shen, J. J. Xia, and P.-T. Yap, "Diverse data augmentation for learning image segmentation with cross-modality annotations," *Medical image analysis*, vol. 71, p. 102060, 2021.

[18] W. Wang, X. Yu, B. Fang, D.-Y. Zhao, Y. Chen, W. Wei, and J. Chen, "Cross-modality lge-cmr segmentation using image-to-image translation based data augmentation," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

[19] G. Hermosilla, D.-I. H. Tapia, H. Allende-Cid, G. F. Castro, and E. Vera, "Thermal face generation using stylegan," *IEEE Access*, vol. 9, pp. 80511–80523, 2021.

[20] M. Abdrakhmanova, A. Kuzdeuov, S. Jarju, Y. Khassanov, M. Lewis, and H. A. Varol, "Speakingfaces: A large-scale multimodal dataset of voice commands with visual and thermal video streams," *Sensors*, vol. 21, no. 10, p. 3465, 2021.

[21] Y. Li, Y. Ko, and W. Lee, "A feasibility study on translation of rgb images to thermal images: Development of a machine learning algorithm," *SN Computer Science*, vol. 4, no. 5, p. 555, 2023.

[22] M. Chaubey, L. K. Singh, and M. Gupta, "Estimation of missing video frames using kalman filter," *Multimedia Tools and Applications*, pp. 1–21, 2023.

[23] Y. Li, D. Roblek, and M. Tagliasacchi, "From here to there: Video inbetweening using direct 3d convolutions," *arXiv preprint arXiv:1905.10240*, 2019.

[24] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.

[26] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[27] A. Rangesh, B. Zhang, and M. M. Trivedi, "Gaze preserving cyclegans for eyeglass removal and persistent gaze estimation," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 377–386, 2022.

[28] H. Liu, T. Taniguchi, K. Takenaka, Y. Tanaka, and T. Bando, "Reducing the negative effect of defective data on driving behavior segmentation via a deep sparse autoencoder," 10 2016.

[29] H. Liu, T. Taniguchi, K. Takenaka, and T. Bando, "Defect-repairable latent feature extraction of driving behavior via a deep sparse autoencoder," *Sensors*, vol. 18, no. 2, 2018.

[30] R. Greer, A. Gopalkrishnan, M. Keskar, and M. M. Trivedi, "Patterns of vehicle lights: Addressing complexities of camera-based vehicle light datasets and metrics," *Pattern Recognition Letters*, 2024.

[31] A. Doshi and M. M. Trivedi, "Examining the impact of driving style on the predictability and responsiveness of the driver: Real-world and simulator analysis," in *2010 IEEE Intelligent Vehicles Symposium*, pp. 232–237, IEEE, 2010.

[32] A. Tawari and M. M. Trivedi, "Robust and continuous estimation of driver gaze zone by dynamic analysis of multiple face videos," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 344–349, IEEE, 2014.

Driver Activity Classification Using Generalizable Representations from Vision-Language Models

Ross Greer * regreer@ucsd.edu
Mathias V. Andersen * mvan19@student.aau.dk
Andreas Møgelmose anmo@create.aau.dk
Mohan Trivedi mtrivedi@ucsd.edu

Abstract

Driver activity classification is crucial for ensuring road safety, with applications ranging from driver assistance systems to autonomous vehicle control transitions. In this paper, we present a novel approach leveraging generalizable representations from vision-language models for driver activity classification. Our method employs a Semantic Representation Late Fusion Neural Network (SRLF-Net) to process synchronized video frames from multiple perspectives. Each frame is encoded using a pretrained vision-language encoder, and the resulting embeddings are fused to generate class probability predictions. By leveraging contrastively-learned vision-language representations, our approach achieves robust performance across diverse driver activities. We evaluate our method on the Naturalistic Driving Action Recognition Dataset, demonstrating strong accuracy across many classes. Our results suggest that vision-language representations offer a promising avenue for driver monitoring systems, providing both accuracy and interpretability through natural language descriptors. We make our code available at https://github.com/viborgen/Driver-Activity-Classification-Using-Generalizable-Representations-from-Vision-Language-Models

1. Introduction

Distracted driving is a common factor in many vehicle accidents [1]. Systems which monitor the driver can offer advisories to the driver which encourage maintained focus on the road [2–7]. These advisories can be effective at reducing the occurrence or severity of related accidents [8].

Another solution to individual transportation lies in autonomous vehicles, with a distant goal that distracted driving is no longer a problem if the person in the driver's seat is not expected to be controlling the vehicle. However, current systems encounter failure cases and novel scenarios [9, 10]. Systems cannot safely transfer control without awareness of the driver, as the driver may be sleeping or pre-occupied with a distracting activity. For this reason, in-cabin driver monitoring and understanding of the driver state is critical for control transitions in autonomous systems too [11–13].

2. Related Research

Models which treat driver monitoring as a closed-set task [14, 15] have found success on benchmark datasets [16, 17].

However, the real environment is open-set [18, 19]. While our provided method is not open-set in its training data, by using a foundation model backbone, the encoding network has already learned representations of nearly any activity class. This makes the method highly adaptable to any visual activity class, though learning to classify those encoded representations may still require closed-set supervised learning (or, an unsupervised or active method to identify novel classes [20–22]) to provide desired predictions suitable to the open-set world.

Further, the real environment contains drivers which are out-of-distribution for a fixed set of training subjects. This is a problem when using data-driven methods which are tuned based on visual features. Some solutions lie in abstractions which remove the driver identity from the image [23, 24]. Related to this problem is the challenge of generalizing to drivers without training data; for most situations, it is infeasible that the vehicle monitoring system capture input of the driver, annotate this input, and use it to finetune a system, especially in the case of non-standard views or sensor rates [25]. This motivates the need for zero-shot learning, where the system is expected to perform with zero prior training instances of the test subject [26, 27].

In this research, we introduce a method which represents the driver in a language-based visual descriptor. Though this representation utilizes image-based features, the features are learned in relationship to verbal descriptors, which pushes the representation from one based purely on pixel values to a representation which is based on the meaning of patterns found in those pixels, to the extent that they can be

*Authors contributed equally. R. Greer, M. V. Andersen, and M. Trivedi are with the Laboratory for Intelligent & Safe Automobiles at the University of California San Diego. A. Møgelmose is with the Visual Analysis and Perception Lab at Aalborg University.

described by natural language.

3. Methodology

3.1. Algorithm

Our algorithm for driver activity classification is presented in Algorithm 1, including encoding, network approximation, and post-processing.

Algorithm 1: SRLF Activity Classification Algorithm

Input: Synchronized video frames
Output: Filtered probabilities per instance
foreach *triplet of frames* **do**
  foreach *frame* **do**
    Create an embedding for the image using the CLIP pretrained vision encoder;
  Pass the three embeddings as input to the SRLF neural network;
  Take argmax over output to receive single class probability per frame;
  Apply a mode filter with window size w over the resulting probabilities;

We use $w = 141$ for our inference data sampled at 30 Hz, but this parameter should be tuned to match the typical duration of the driver activities, relative to the rate at which the network generates predictions or processes input.

3.2. Semantic Representation Late Fusion Neural Network

Our network, Semantic Representation Late Fusion Neural Network (SRLF-Net) is presented in Figure 1. The network consists of $N = 3$ parallel CLIP ViT image encoders, followed each by an FCN encoder, after which the outputs of the N tracks are fused before entering a deep FCN network to generate class probability output.

3.3. Leveraging Generalizable Representations from Language-Vision Foundation Models

With this representation, rather than the descriptor of each driver being a specific array of pixels which may represent that driver's facial structure, hairstyle, skin color, size, and other non-relevant traits, the information bottleneck and pretraining mechanism instead reduce the amount of information and preserve (at least, to the ability of the optimizer) only features which are useful in organizing the images in a latent space that is separable by language. Of course, it is possible that with language we can describe concepts like facial structure, hairstyle, skin color, etc., but what is important is that the verbal description of these properties is a much lower amount of information than having the complete set of pixels which define that facial structure or hairstyle or skin color. With this representation, it is our hypothesis that the model becomes significantly more generalizable when trained, as it loses its ability to overfit to the very-individual properties of specific drivers.

Taken to an extreme, we can view the act of classifying an image as a reduction to the minimal number of bits to represent the information we care about from an image. With this in mind, we can view the image itself as the representation with the most information (which may be more than is required to solve the problem, containing both noise and irrelevant detail), and the class itself as the most compact. It is possible to use a large language model to directly output a prediction of a class, but this relies on the tuning of many components, in particular, the text encoder of the classification-request prompt and the associated prompt phrasing, and the ability of the model which learns to decode the image to a class according to this prompt. In our results, we show that current large language models struggle to learn this task satisfactorily. Because the image encoding representation is a less-reduced representation, we suggest that this can be used as an intermediate (not too large, not too small) representation of the relevant information, from which we can learn appropriate patterns without requiring the tuning of a text encoder or the finetuning of the model parameters which connect a prompt encoding and vision encoding, significantly reducing computational and data requirements while still maintaining the necessary level of information to solve the classification problem.

3.4. Separating Visual and Semantic Information using Order-based Augmentation

While we would ideally extract a semantic-level representation of the images and remove the ability to overfit to pixel configurations, the CLIP representation still carries some image features forward. However, we introduce a method which mitigates the overfitting possibility by a specialized data augmentation. If we treat the order in which the three views are passed to the network as random, we may be able to push the model to learn features within the 768-vector which represent semantic information as opposed to image-feature information, since the image-feature information would vary for each view while the semantic feature information should remain consistent. This method can also be extended to any number of views. While an overparameterized model may simply learn additional representations (for each permutation of image order), an appropriately-parameterized model may show better generalizability performance through this semantic-pass-filter information bottleneck.

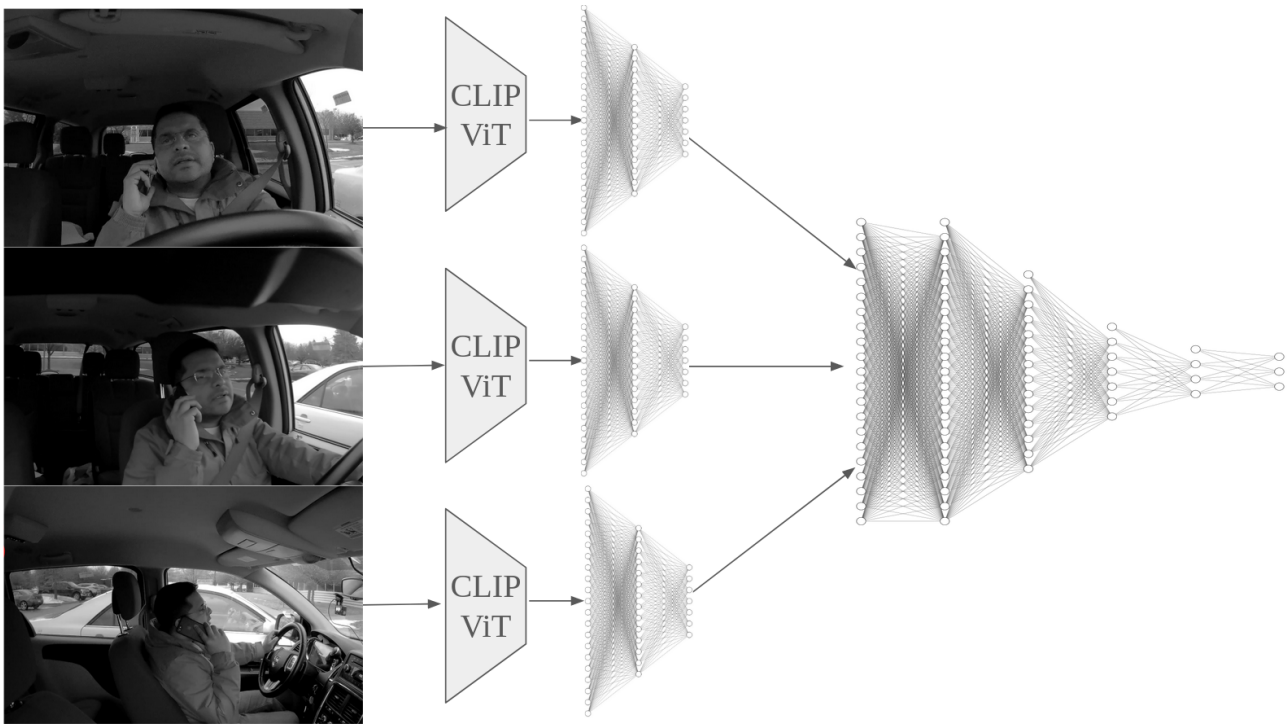


Figure 1. The Semantic Representation Late Fusion Network (SRLF-Net) takes images from multiple perspectives as input. Each image is sent to a CLIP encoder. Our experiments use the Vision Transformer backbone, base size, with size 32 patches. These representations are then further encoded using independent (non-shared-weight) fully-connected layers, each followed by batch normalization, ReLU activation, and dropout (rates 0.5 and 0.6 respectively). We use input size 768, and use two layers, compressing once to 512 and then to 256. These representations are then concatenated and used as input to another series of fully-connected layers (fusion step), again using batch normalization and ReLU activation between each. The size of these layers are 768, 768, 512, 256, 128, then n (number of classes), which is 16 for our experiments. A residual connection is added between the concatenation and the second layer sized 768.

4. Experimental Evaluation

4.1. Dataset

We utilize the Naturalistic Driving Action Recognition Dataset from the AI City Challenge [28], which consists of approximately 62 hours of footage, acquired from 69 participants. Each participant performed 16 different tasks, including but not limited to telephonic conversations, eating, and reaching backward, in a randomized order, as specified in Table 1.

The data includes three camera positions installed within a vehicle, as in Figure 2, positioned to capture from varied angles and synchronized to record simultaneously. The data collection was executed in two phases for each participant: the first without any visual obstructions and the second incorporating visual obstructions to appearance (e.g., sunglasses, hats). Thus, six videos were collected per participant—three from the non-obstructed phase and three from the obstructed phase.

Class	Activity Label	Dist. %
0	Normal Forward Driving	59.01
1	Drinking	1.49
2	Phone Call(right)	2.78
3	Phone Call(left)	2.97
4	Eating	3.29
5	Text (Right)	3.44
6	Text (Left)	3.56
7	Reaching behind	1.40
8	Adjust control panel	2.42
9	Pick up from floor (Driver)	1.31
10	Pick up from floor (Passenger)	2.15
11	Talk to passenger at the right	3.52
12	Talk to passenger at backseat	3.46
13	Yawning	1.87
14	Hand on head	3.45
15	Singing or dancing with music	3.85

Table 1. Table of Driver Activity Classifications.

Figure 2. Illustration of multi-perspective in-cabin camera views for monitoring driver behavior under the class '0: Normal Forward Driving'. (1) Dashboard view. (2) Rear-view. (3) Side view.

4.2. Training Details

We detail our evaluation data splits in the following sections, with care to have images of individuals binned only to one set out of training and test. We divide our training set into two groups; 80% to train and 20% to validation, with possible overlap in individuals (though no same frames are shared). With our training set, we train SRLF-Net for up to 100 epochs, employing early stopping on a validation loss criteria. We use the adam optimizer (learning rate of 0.0001), 1cycle learning rate schedule policy [29], and cross-entropy loss.

For testing, we utilize the 7-fold data split provided in the dataset, dividing into 7 near-even groups of participants. This allows us to approximate generalizability with a 7-fold average.

4.3. Evaluation Over All Classes

The results for 7-fold test are seen in Table 2. We achieve an average accuracy of 71.64 %, showcasing the promising use of the method, notable in comparison to 6.25% expected accuracy of random selection for sixteen classes.

K-fold	Accuracy
1	68.09 %
2	74.40 %
3	73.60 %
4	71.37 %
5	70.15 %
6	75.34 %
7	68.53 %
Average:	71.64
Standard Deviation:	2.88

Table 2. Table of k-fold cross-validation accuracies and average accuracy.

As illustrated in Figure 3, the model observes a large favorability for class 0 (Normal Forward Driving) likely due

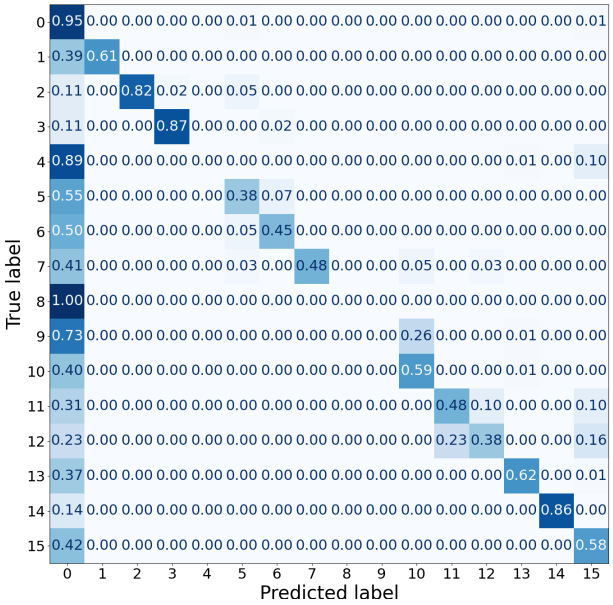


Figure 3. Confusion matrix for best performing k-fold 6 including a mode filter, resulting in a performance of 77.10 %.

to the skewed distributions of the data, as portrayed in Table 1, with phone call and hand-on-head the next most-correctly-classified classes. Adjusting the control panel shows the most confusion with the default driving class. Straight forward driving accounts for 59.01 % of the data, resonating binary test to differentiate between straight forward driving and all other classes in Figure 4. For more accurate classification, it would be beneficial to mitigate the effects of the confounding majority class ("normal driving"); we explore experiments in class-weighting, but find these effects to not be strong enough to counter the adverse learning effect. As another solution, we consider the use of an early-stage binary classifier to separate normal driving from distracted driving. The binary classifier is imperfect (as shown in Figure 4, and in the next section, we carry out an additional distraction-classification experiment excluding the "normal driving" class, on the assumption that some strong binary classifier may be achieved with further architectural exploration.

4.4. Distracting Activities Only: Evaluating Without Normal Driving Class

Our architecture, in combination with a dataset heavily skewed towards normal driving, tends to overpredict the normal driving class. To understand how well the model separates between the distracting activity classes, we run an experiment by which we assume there is some "perfect" binary classifier which can distinguish between normal driving and distracted driving, and then use our model to classify only between these distraction classes. The results of

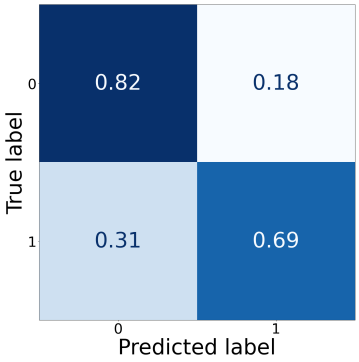


Figure 4. Binary Confusion matrix for best performing k-fold 6 only including class 0 for straight forward driving and a combination of all other activity classes, performing 77.22 % accuracy.

this experiment are illustrated in Figure 5. The model, in general, predicts the correct class with the greatest likelihood for any given activity class, though for some individual classes, this likelihood may be less than >50%. Phone call and hand-on-head again show the best performance.

We also highlight the importance of the mode-filter post-processing step; without the mode filter, the accuracy is 63.66%, and with the mode filter, this accuracy rises to 70.06%. This filter leverages the knowledge that there is a certain rate at which a driver can reasonably change between tasks (i.e. it would be unexpected for a driver to oscillate between different distracting activities at 30 Hz, even if the camera captures and model infers at that rate).

5. Concluding Remarks and Future Research

To begin, we highlight some recommended opportunities for future research:

1. Comparison to text-encoding methods, such as vector products between text and image encodings, or even the evaluation of prompted vision-language systems to determine classes of images. We note that we have began a series of experiments using LLaVA, but the computation time on such methods *significantly* exceeds the method shown in this paper, without offering stronger preliminary results. In relation to these methods, our presented algorithm does carry the benefit of immediate applicability to multiple simultaneous views.

2. The integration of temporal information (either as post-processing, or addition of LSTM or Transformer models early in the architecture) may be very useful, since driver activities occur over time, with valuable information in these action dynamics.

3. Evaluation on combinations of non-consistent views. It would be interesting to merge multiple datasets which share some classes in common, so that we can evaluate generalizability to further views and subjects.

4. Integration into open-set novelty detection methods, such that the system can expand its number of classes, retraining if necessary, when new activities are introduced.

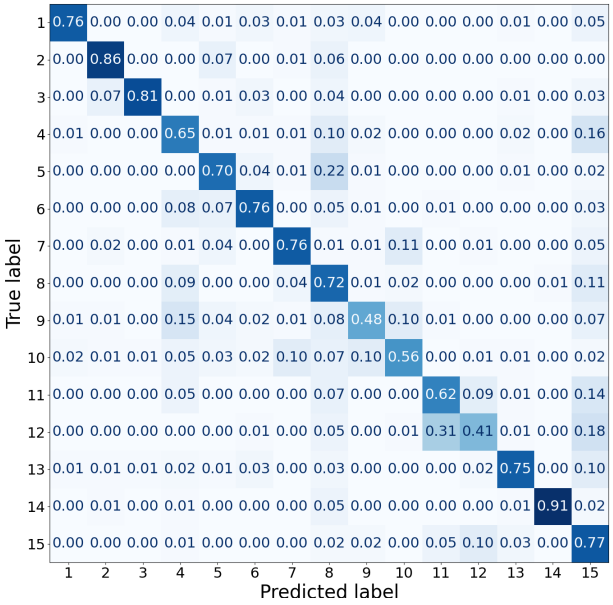


Figure 5. Confusion matrix for best performing k-fold 6 without class 0 for straight forward driving and including a mode filter, performing 70.06% accuracy. By removing the forward driving class, the accuracy metric decreases slightly (simply because the over-predicted forward driving class accounted for a majority of the dataset), but the average performance over classes actually increases from 50.44% to 70.13%. The alignment of average per-class accuracy and overall accuracy is a strong indicator of the model's effective learning.

In this research, we present a new perspective of the vision-language contrastively-learned encoding as a fundamental new representation of an image, which contains both visual information as well as semantic information. We show that from this information, it is possible to classify driver activity into a variety of distraction classes with fairly strong accuracy, and further, that our algorithm can adapt to any number of simultaneous views. Vision-language models may lead to driver monitoring systems which are more accurate, robust, and generalizable; suitable for an open-set of possible distractions; and directly explainable [30] via language.

References

[1] Ahmed Sajid Hasan, Mohammad Jalayer, Eric Heitmann, and Joseph Weiss. Distracted driving crashes: a review on data collection, analysis, and crash prevention methods. *Transportation research record*, 2676(8):423–434, 2022. 1

[2] Sean L Gallahan, Ghilan F Golzar, Abhishek P Jain, Ashley E Samay, Tyler J Trerotola, John G Weisskopf, and Nathan Lau. Detecting and mitigating driver distraction with motion capture technology: Distracted driving warning system. In *2013 IEEE Systems and Information Engineering Design Symposium*, pages 76–81. IEEE, 2013. 1

[3] Ross Greer, Lulua Rakla, Anish Gopalan, and Mohan Trivedi. (safe) smart hands: Hand activity analysis and distraction alerts using a multi-camera framework. *arXiv preprint arXiv:2301.05838*, 2023. 1

[4] Ross Greer and Mohan Trivedi. Ensemble learning for fusion of multiview vision with occlusion and missing information: Framework and evaluations with real-world data and applications in driver hand activity recognition. *arXiv preprint arXiv:2301.12592*, 2023. 1

[5] S Gobhinath, V Aparna, and R Azhagunacchiya. An automatic driver drowsiness alert system by using gsm. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pages 125–128. IEEE, 2017. 1

[6] Ross Greer, Lulua Rakla, Samveed Desai, Afnan Alofi, Akshay Gopalkrishnan, and Mohan Trivedi. Champ: Crowdsourced, history-based advisory of mapped pedestrians for safer driver assistance systems. *arXiv preprint arXiv:2301.05842*, 2023. 1

[7] Ross Greer, Samveed Desai, Lulua Rakla, Akshay Gopalkrishnan, Afnan Alofi, and Mohan Trivedi. Pedestrian behavior maps for safety advisories: Champ framework and real-world data analysis. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. IEEE, 2023. 1

[8] Susann Winkler, Juela Kazazi, and Mark Vollrath. Distractive or supportive–how warnings in the head-up display affect drivers' gaze and driving behavior. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 1035–1040. IEEE, 2015. 1

[9] Ross Greer and Mohan Trivedi. Towards explainable, safe autonomous driving with language embeddings for novelty identification and active learning: Framework and experimental analysis with real-world data sets. *arXiv preprint arXiv:2402.07320*, 2024. 1

[10] Ernestine Fu, David Hyde, Srinath Sibi, Mishel Johns, Martin Fischer, and David Sirkin. Assessing the effects of failure alerts on transitions of control from autonomous driving systems. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1956–1963. IEEE, 2020. 1

[11] Akshay Rangesh, Nachiket Deo, Ross Greer, Pujitha Gunaratne, and Mohan M Trivedi. Autonomous vehicles that alert humans to take-over controls: Modeling with real-world data. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 231–236. IEEE, 2021. 1

[12] Akshay Rangesh, Nachiket Deo, Ross Greer, Pujitha Gunaratne, and Mohan M Trivedi. Predicting take-over time for autonomous driving with real-world data: Robust data augmentation, models, and evaluation. *arXiv preprint arXiv:2107.12932*, 2021. 1

[13] Ross Greer, Nachiket Deo, Akshay Rangesh, Pujitha Gunaratne, and Mohan Trivedi. Safe control transitions: Machine vision based observable readiness index and data-driven takeover time prediction. *arXiv preprint arXiv:2301.05805*, 2023. 1

[14] Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen. Transdarc: Transformer-based driver activity recognition with latent space feature calibration. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 278–285. IEEE, 2022. 1

[15] Alina Roitberg, Kunyu Peng, Zdravko Marinov, Constantin Seibold, David Schneider, and Rainer Stiefelhagen. A comparative analysis of decision-level fusion for multimodal driver behaviour understanding. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1438–1444. IEEE, 2022. 1

[16] Manuel Martin, Alina Roitberg, Monica Haurilet, Matthias Horne, Simon Reiß, Michael Voit, and Rainer Stiefelhagen. Drive&act: A multi-modal dataset for fine-grained driver behavior recognition in autonomous vehicles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2801–2810, 2019. 1

[17] Alina Roitberg, Monica Haurilet, Manuel Martinez, and Rainer Stiefelhagen. Uncertainty-sensitive activity recognition: A reliability benchmark and the caring models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 3814–3821. IEEE, 2021. 1

[18] Alina Roitberg, Chaoxiang Ma, Monica Haurilet, and Rainer Stiefelhagen. Open set driver activity recognition. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1048–1053. IEEE, 2020. 1

[19] Kunyu Peng, Cheng Yin, Junwei Zheng, Ruiping Liu, David Schneider, Jiaming Zhang, Kailun Yang, M Saquib Sarfraz, Rainer Stiefelhagen, and Alina Roitberg. Navigating open set scenarios for skeleton-based action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4487–4496, 2024. 1

[20] Alina Roitberg, Ziad Al-Halah, and Rainer Stiefelhagen. Informed democracy: voting-based novelty detection for action recognition. *arXiv preprint arXiv:1810.12819*, 2018. 1

[21] Ross Greer, Bjørk Antoniussen, Andreas Møgelmose, and Mohan Trivedi. Language-driven active learning for diverse open-set 3d object detection, 2024. 1

[22] Ross Greer, Bjørk Antoniussen, Mathias V. Andersen, Andreas Møgelmose, and Mohan M. Trivedi. The why, when, and how to use active learning in large-data-driven 3d object detection for safe autonomous driving: An empirical exploration, 2024. 1

[23] Kunyu Peng, Alina Roitberg, Kailun Yang, Jiaming Zhang, and Rainer Stiefelhagen. Delving deep into one-shot

skeleton-based action recognition with diverse occlusions. *IEEE Transactions on Multimedia*, 2023. 1

[24] Surendrabikram Thapa, Julie Cook, and Abhijit Sarkar. Face de-identification of drivers from nds data and its effectiveness in human factors. 2023. 1

[25] Mathias Viborg Andersen, Ross Greer, Andreas Møgelmose, and Mohan Trivedi. Learning to find missing video frames with synthetic data augmentation: A general framework and application in generating thermal images using rgb cameras, 2024. 1

[26] Alina Roitberg, Manuel Martinez, Monica Haurilet, and Rainer Stiefelhagen. Towards a fair evaluation of zero-shot action recognition using external data. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[27] Simon Reiß, Alina Roitberg, Monica Haurilet, and Rainer Stiefelhagen. Activity-aware attributes for zero-shot driver behavior recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 902–903, 2020. 1

[28] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th ai city challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023. 3

[29] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 4

[30] Akash Sonth, Abhijit Sarkar, Hirva Bhagat, and Lynn Abbott. Explainable driver activity recognition using video transformer in highly automated vehicle. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. IEEE, 2023. 5

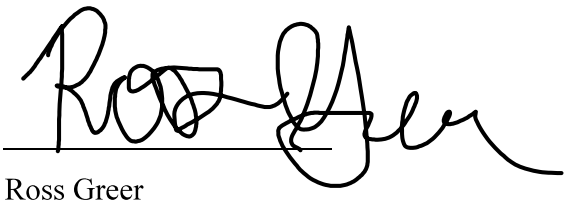
Co-Author Permission Letter

10/5/2024

Mathias Viborg Andersen has my permission to include material that has been submitted for publication, of which I was a co-author.

Andersen, Mathias Viborg; Greer, Ross; Møgelmose, Andreas; and Trivedi, Mohan M. "Learning to Find Missing Video Frames with Synthetic Data Augmentation: A General Framework and Application in Generating Thermal Images Using RGB Cameras," arXiv preprint arXiv:2403.00196.

Greer, Ross; Andersen, Mathias Viborg; Møgelmose, Andreas; and Trivedi, Mohan M. "Driver Activity Classification Using Generalizable Representations from Vision-Language Models," arXiv preprint arXiv:2404.14906.



10 May 2024

Ross Greer