Driver Drowsiness

Ray Hung, Mayank Rai, Kirti Prakash

https://github.com/kp-commit/mids-207-group2-final-project/tree/main

Table of Contents

Dataset

Driver Drowsiness Dataset

- Derived from the Real-Life Drowsiness Dataset, focusing on drivers' faces.
- Frames were extracted from videos using VLC software to create images.
- The *Viola-Jones algorithm* was applied to isolate regions of interest in the captured images.
- Initially used for training and testing Convolutional Neural Networks (CNNs) in a research paper titled "Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques."

Source: Kaggle (based on paper: https://doi.org/10.1007/978-981-33-6893-4_6)

Motivation

Driver drowsiness is the leading reason for a large number of road accidents in the United States.

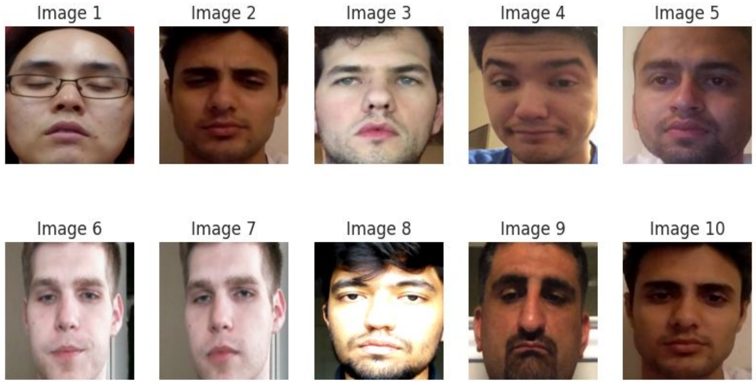
Our objective is to explore different CNN techniques with the aim of constructing a model capable of generalizing to unfamiliar images in efforts to improve public safety.

EDA

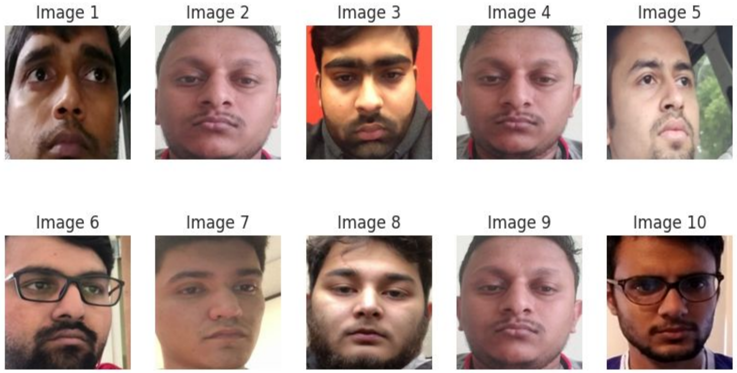
Images

total .png images: 41,793 Image size: **227 x 227**

Drowsy: **22,348**



Non Drowsy: **19,445**



Issues with Image dataset

Karolinska Sleepiness Scale (KSS)

Extremely alert	1
Very alert	2
Alert	3
Rather alert	4
Neither alert nor sleepy	5
Some signs of sleepiness	6
Sleepy, but no effort to keep awake	7
Sleepy, but some effort to keep awake	8
Very sleepy, great effort to keep awake, fighting sleep	9
Extremely sleepy, can't keep awake	10

True Positive False Positive

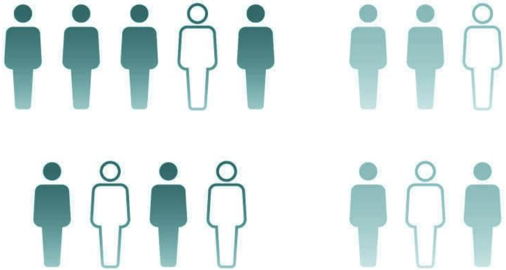
Drowsy

Non Drowsy

False Negative **True Negative**

- The original dataset included **False Positive** and **False Negative labels**.
- **Stratified sampling** approach was adopted to address this problem by maintaining a balanced representation of these categories.

Stratified Sampling



Identified **26 persons in dataset.** Approx, **1,000 images per person.** This varies with some less resulting in a lot of noise.

Reduced data per person with sampling high quality images. Propose **10 images per person = 260 * 2 classes** with self annotation.

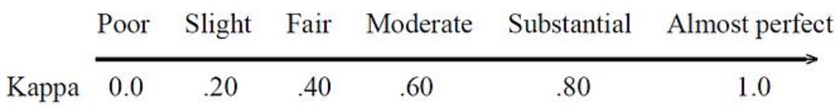
150 random images each were selected from directories labeled **'Drowsy'** and **Non-Drowsy**' to correct for **class imbalance** and then annotated with a label of 0 (NonDrowsy) or 1 (Drowsy)

Inter-annotator agreement (IAA)

Image	Kirti Review	Mayank Review	Ray Review	Fleiss' kappa
A0228.png	0.95	0.93	0.97	0.95
A0587.png	0.85	0.83	0.87	0.85
A0632.png	0.2	0.18	0.22	0.2
A0650.png	0.95	0.93	0.97	0.95

Interpreting IAA and IAR scores is not straightforward, as different annotation tasks may have varying levels of difficulty, ambiguity, subjectivity, or variability. Generally speaking, scores above 0.8 or 0.9 indicate high agreement or reliability.

Interpretation of Kappa



Kappa	Agreement
< 0	Less than chance agreement
0.01–0.20	Slight agreement
0.21– 0.40	Fair agreement
0.41–0.60	Moderate agreement
0.61–0.80	Substantial agreement
0.81–0.99	Almost perfect agreement

Kappa for **Drowsy**: **0.91**

Kappa for **Non Drowsy: 0.69**

Data Processing Methodology

Train/Test Split

- To account for imbalances, our team manually annotated **510 images** and then divided into separate subsets for training, validation, and testing.
- Due to the labeling process, there is a potential overlap where the test subset may contain images of individuals encountered during model training.
- To evaluate the model's performance on previously unseen data, a distinct set of images featuring individuals with names starting from letters **S through Z** has been separated. This separation is intended to assess the model's ability to generalize to new data.

Training (60%)

Validation (10%)

Test (30%)

Drowsy (Letters A-R)

Drowsy (Letters S-Z)

Non-Drowsy (Letters A-R)

Non-Drowsy (Letters S-Z)

Used only for final evaluation

Data processing

Train

Drowsy (206)
+
Non-Drowsy (164)

Step 1

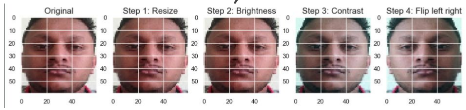
Randomly select 150 training samples / class Drowsy(1)/NonDrowsy(0)

Balanced data selection.

Removes Biasness

Step 2

Image resizing (80x80)
Normalization
Train/Val Split
Augmentation



Reduces model memory footprint.

Improves generalization

Train/Validation

Test

Drowsy (70)
+
Non-Drowsy (70)

Image resize(80x80)

Prediction/ Evaluation

Model Training Methodology

Approach to Model Training

CNN Model V1: Full Face

A baseline CNN model was trained using **3 convolution filters** on **full face images**

↓

CNN Model V2: Full Face

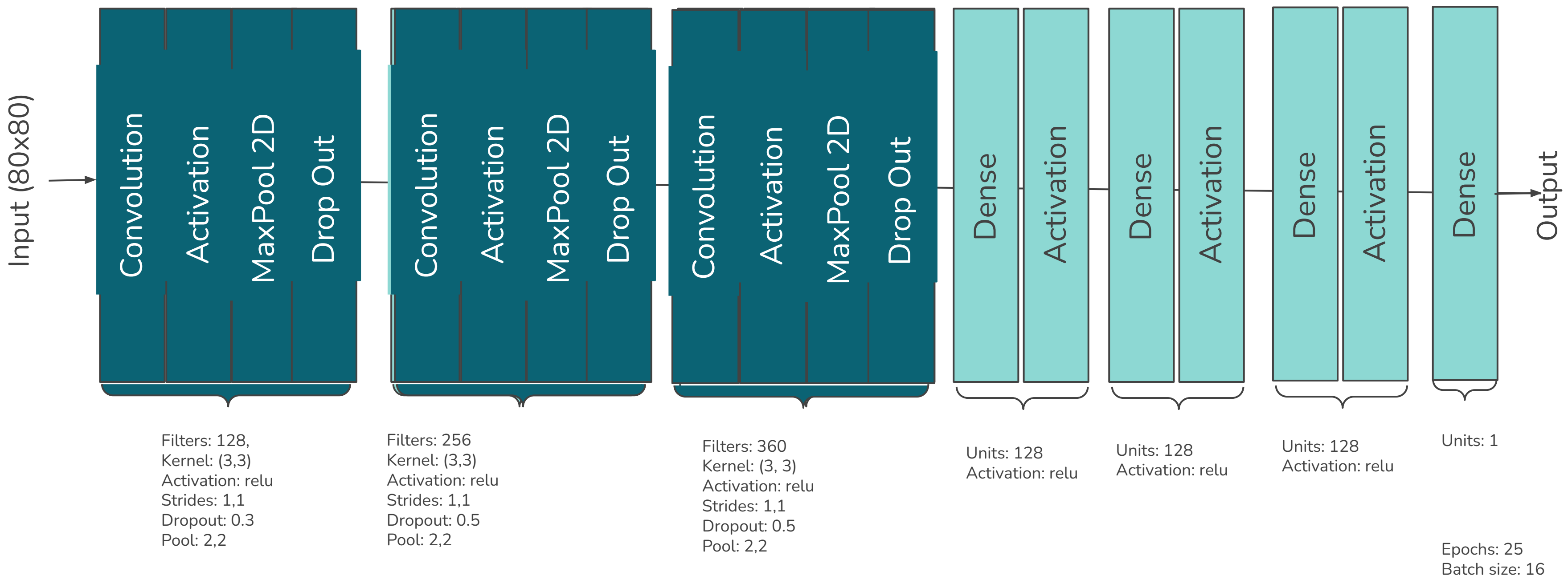
Trained with using **2 convolution filters** on **full face images**

↓

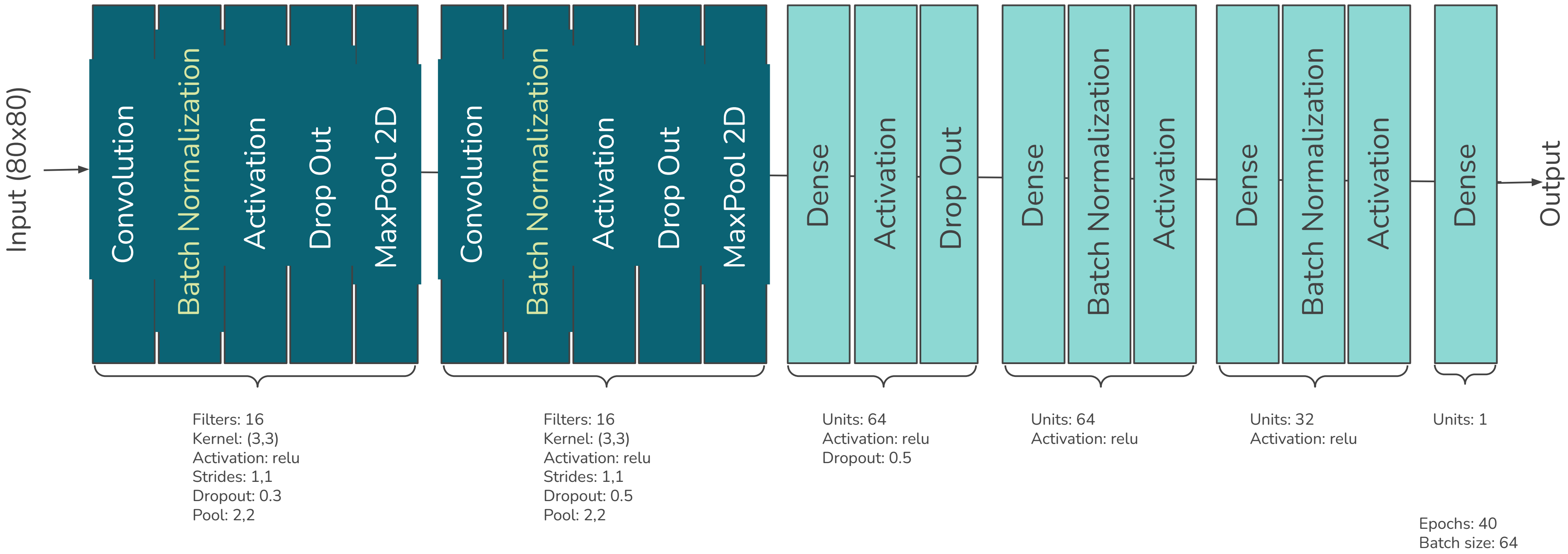
CNN Model V3: Eye

Trained using **2 convolution filters** using **extracted eye images**

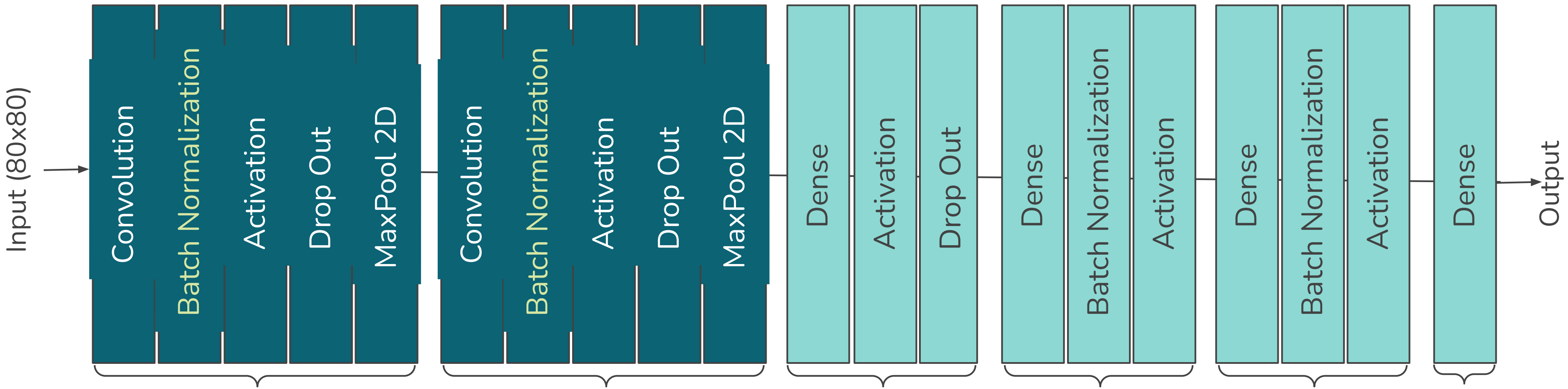
Baseline Model CNN v1: full face



Model CNN v2: full face



Model CNN v3: eye



Filters: 8
Kernel: (3,3)
Activation: relu
Strides: 1,1
Dropout: 0.3
Pool: 2,2

Filters: 16
Kernel: (3,3)
Activation: relu
Strides: 1,1
Dropout: 0.5
Pool: 2,2

Units: 128
Activation: relu
Dropout: 0.5

Units: 64
Activation: relu

Units: 32
Activation: relu

Units: 1

Epochs: 30
Batch size: 32

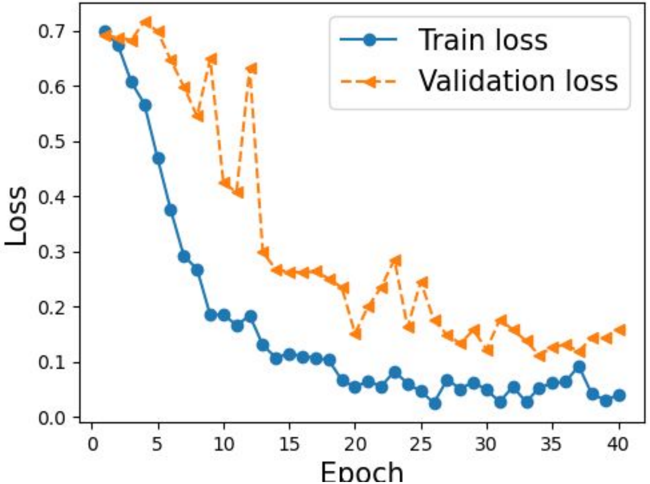
Tuning and Improvements

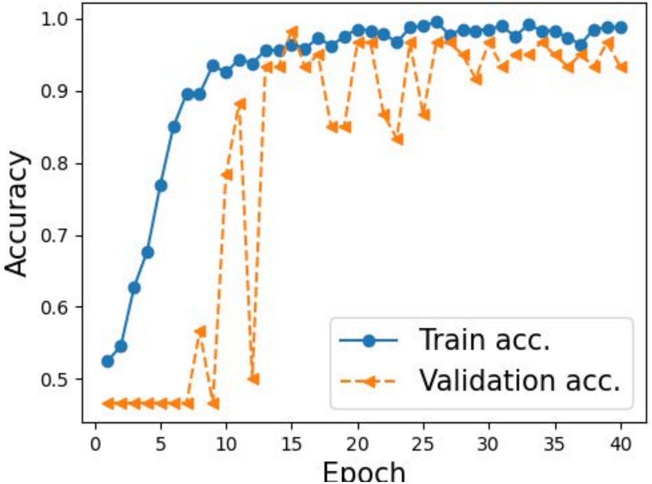
Hyperparameter Tuning

	CNN V1: Full Face	CNN V2: Full Face	CNN V2: Eye Extractor
Filters	[8,8], [8, 16], [16, 8], [16, 16]	[8,8], [8, 16], [16, 8], [16, 16]	[8,8], [8, 16], [16, 8], [16, 16]
Optimizers	Adam, Adagrad, SGD	Adam	Adam
Loss	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy
Kernel Sizes	(3,3), (5,5)	(3,3), (5,5)	(3,3), (5,5)
Pool Sizes	(2,2)	(2,2)	(2,2)
Learning Rate	0.001, 0.011, 0.016	0.001	0.001
Batch Size	16, 32, 64	16, 32, 64	16, 32, 64
Epochs	30, 40	30, 40	30, 40

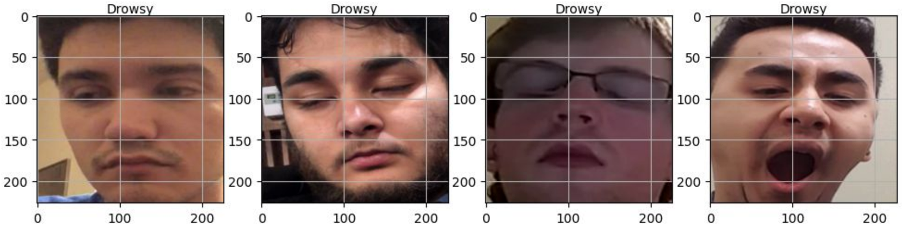
CNN Model V1: Best “Baseline” Model

Learning Rate	Filters	Kernel Size	Strides	optimizer	Conv Dropout Rate	FC Dropout Rate	Units	Pool Size	Epochs	Batch Size	Test Accuracy	Model Size
0.001	[16, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	16	89.99999762	0.824131012



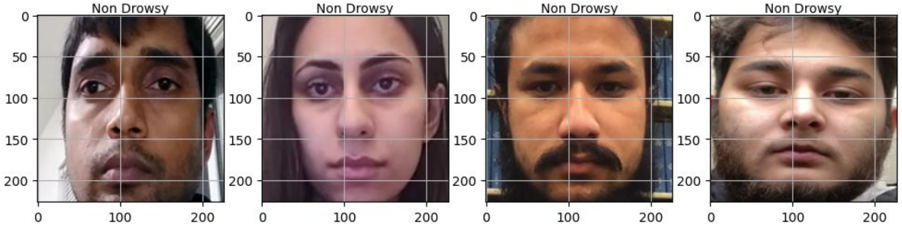


CNN V2:Full Face Image Inputs



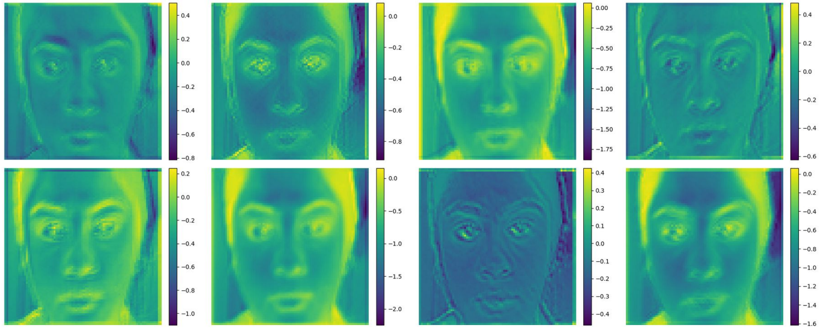
Drowsy

Non-Drowsy

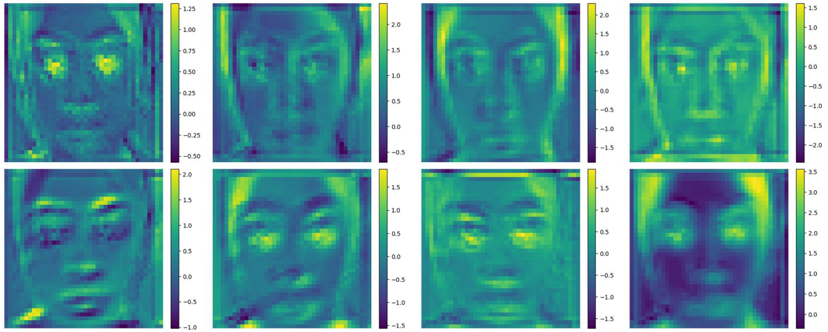


CNN V2:Convolution Filters

Layer 1



Layer 2



Used in CNN to extract features and detect patterns with input data for full face images.

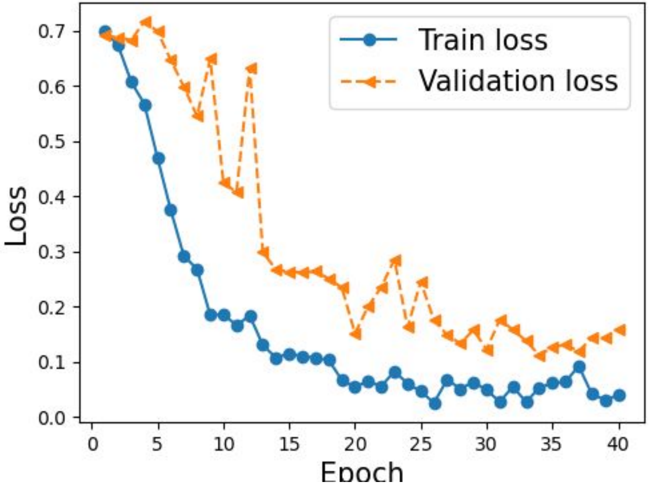
Adding additional layers increase feature diversity

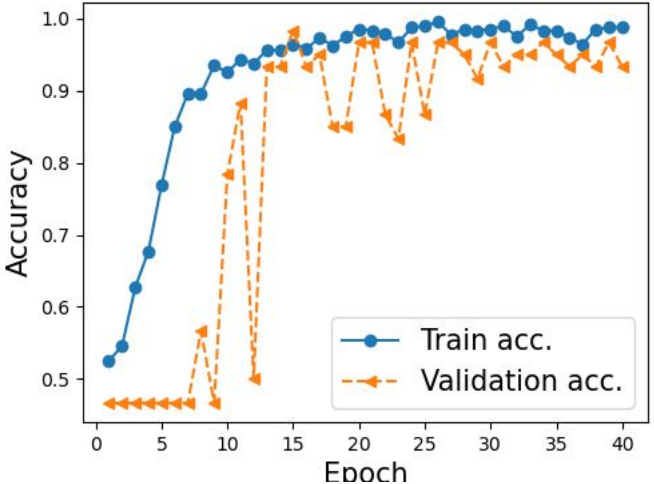
CNN V2:Hyperparameter Tuning

Learning Rate	Filters	Kernel Size	Strides	optimizer	Conv Dropout Rate	FC Dropout Rate	Units	Pool Size	Epochs	Batch Size	Test Accuracy	Model Size	History Plot
0.001	[16, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	16	89.99999762	0.824131012	./plots/full_face/history_34.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	32	87.00000048	3.196109772	./plots/full_face/history_2.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	16	86.00000143	3.196109772	./plots/full_face/history_4.png
0.001	[8, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	64	86.00000143	3.181461334	./plots/full_face/history_51.png
0.001	[16, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	83.99999738	1.621250153	./plots/full_face/history_29.png
0.001	[16, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	32	83.99999738	0.813388824	./plots/full_face/history_47.png
0.001	[8, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	32	81.99999928	0.810214996	./plots/full_face/history_92.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	16	81.00000024	3.196109772	./plots/full_face/history_1.png
0.001	[16, 16]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	32	81.00000024	1.599185944	./plots/full_face/history_23.png
0.001	[16, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	32	81.00000024	0.813388824	./plots/full_face/history_44.png
0.001	[8, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	32	81.00000024	3.181461334	./plots/full_face/history_50.png
0.001	[16, 16]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	80.00000119	3.177555084	./plots/full_face/history_17.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	64	79.00000215	3.196109772	./plots/full_face/history_3.png
0.001	[8, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	16	79.00000215	3.181461334	./plots/full_face/history_52.png
0.001	[8, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	16	79.00000215	1.603092194	./plots/full_face/history_55.png

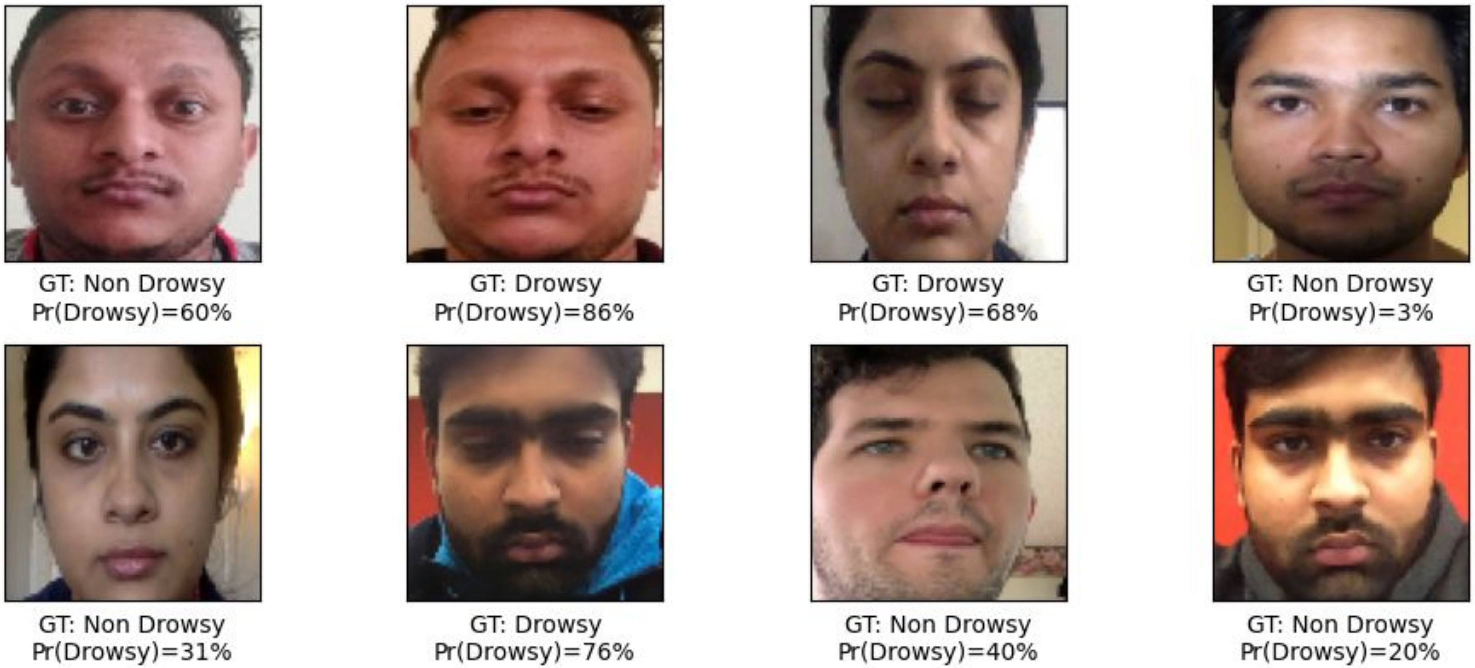
CNN V2: Best Model

Learning Rate	Filters	Kernel Size	Strides	optimizer	Conv Dropout Rate	FC Dropout Rate	Units	Pool Size	Epochs	Batch Size	Test Accuracy	Model Size
0.001	[16, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	16	89.99999762	0.824131012

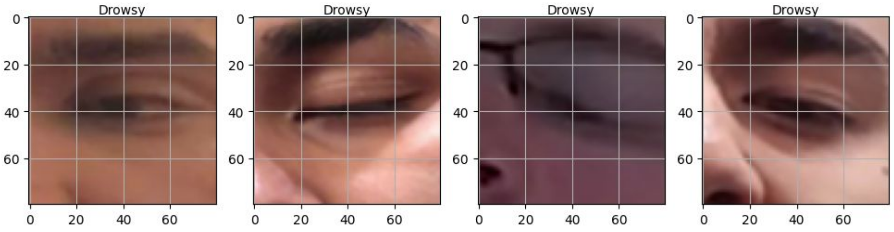




CNN V2: Full Face Model Predictions

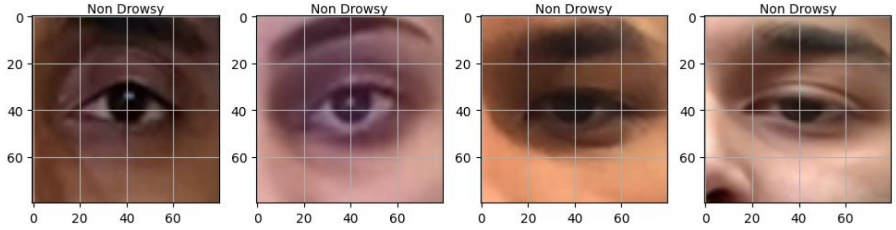


CNN V3: Eye Extractor Model Inputs



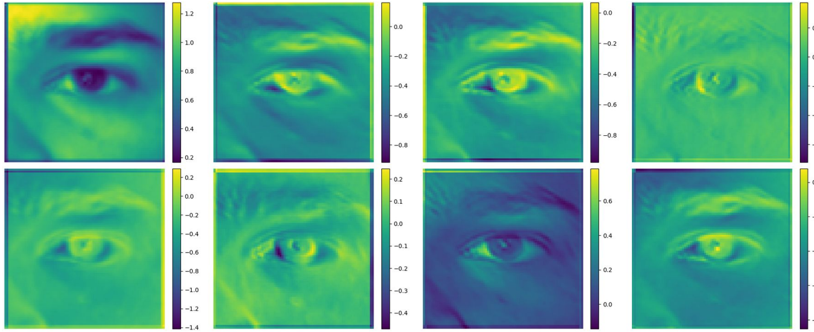
Drowsy

Non-Drowsy

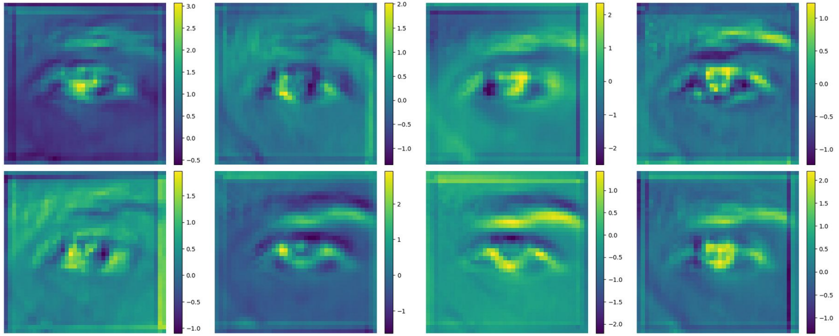


CNN V3: Convolution Filters

Layer 1



Layer 2

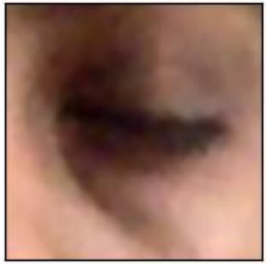


Applied the convolution filters to extract features and patterns from the eye extractor input data

CNN V3: Eye Model Predictions

GT: Non Drowsy
Pr(Drowsy)=83%

GT: Drowsy
Pr(Drowsy)=98%

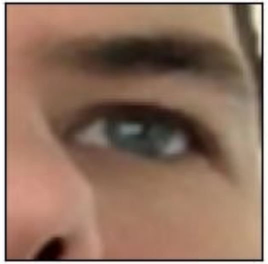


GT: Drowsy
Pr(Drowsy)=94%

GT: Non Drowsy
Pr(Drowsy)=12%

GT: Non Drowsy
Pr(Drowsy)=11%

GT: Drowsy
Pr(Drowsy)=99%



GT: Non Drowsy
Pr(Drowsy)=17%

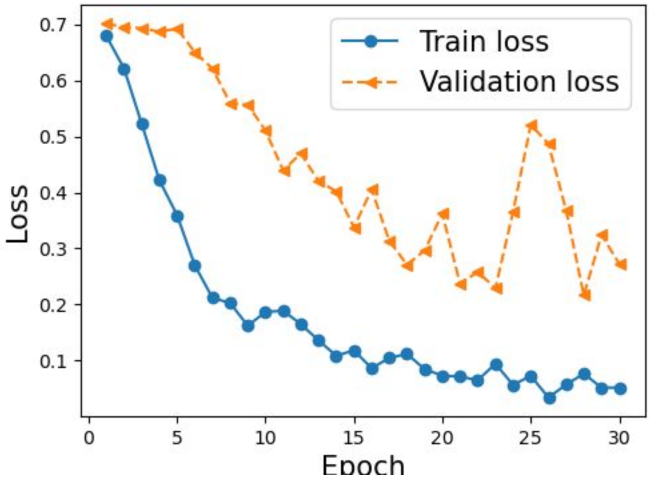
GT: Non Drowsy
Pr(Drowsy)=29%

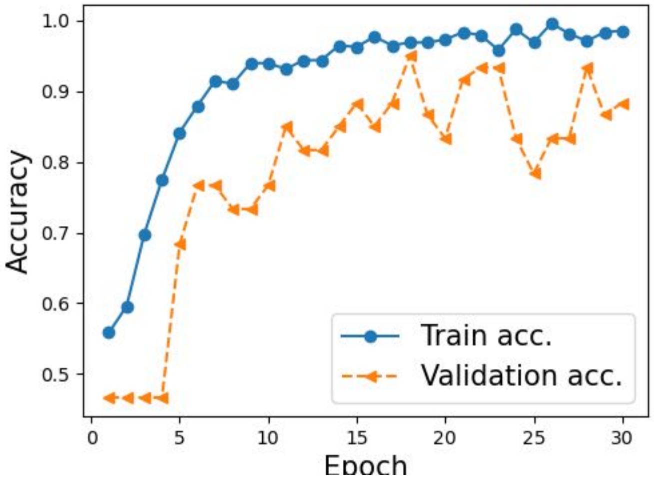
CNN V3: Hyperparameter tuning

Learning Rate	Filters	Kernel Size	Strides	optimizer	Conv Dropout Rate	FC Dropout Rate	Units	Pool Size	Epochs	Batch Size	Test Accuracy	Model Size	History Plot
0.001	[8, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	16	97.00000286	0.81558609	./plots/eye/history_79.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	40	32	95.99999785	1.617740631	./plots/eye/history_11.png
0.001	[8, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	93.99999976	1.612705231	./plots/eye/history_77.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	32	93.00000072	1.617740631	./plots/eye/history_8.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	64	93.00000072	3.196109772	./plots/eye/history_3.png
0.001	[8, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	92.00000167	1.607334137	./plots/eye/history_89.png
0.001	[8, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	32	92.00000167	0.810214996	./plots/eye/history_92.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	92.00000167	3.196109772	./plots/eye/history_5.png
0.001	[16, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	16	91.00000262	1.621250153	./plots/eye/history_28.png
0.001	[8, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	30	32	91.00000262	1.607334137	./plots/eye/history_86.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	64	91.00000262	3.196109772	./plots/eye/history_6.png
0.001	[16, 8]	[(3, 3), (3, 3)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[128, 64, 32]	[(2, 2), (2, 2)]	40	32	89.99999762	1.610507965	./plots/eye/history_41.png
0.001	[16, 16]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	16	87.99999952	1.617740631	./plots/eye/history_7.png

CNN V3: Best Model

Learning Rate	Filters	Kernel Size	Strides	optimizer	Conv Dropout Rate	FC Dropout Rate	Units	Pool Size	Epochs	Batch Size	Test Accuracy	Model Size
0.001	[8, 8]	[(5, 5), (5, 5)]	[(1, 1), (1, 1)]	adam	[0.3, 0.5]	[0.5, 0, 0]	[64, 64, 32]	[(2, 2), (2, 2)]	30	16	97.00000286	0.81558609





Conclusion

Summary: Hyperparameter Tuning

	CNN V1: Full Face	CNN V2: Full Face	CNN V2: Eye Detector
Filters	[8,8], [8, 16], [16, 8], [16, 16]	[8,8], [8, 16], [16, 8], [16, 16]	[8,8], [8, 16], [16, 8], [16, 16]
Optimizers	Adam, Adagrad, SGD	Adam	Adam
Loss	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy
Kernel Sizes	(3,3), (5,5)	(3,3), (5,5)	(3,3), (5,5)
Pool Sizes	(2,2)	(2,2)	(2,2)
Learning Rate	0.001, 0.011, 0.016	0.001	0.001
Batch Size	16, 32, 64	16, 32, 64	16, 32, 64
Epochs	30, 40	30, 40	30, 40

Best Hyperparameters

Model Summary

	CNN V1: Full Face	CNN V2: Full Face	CNN V2: Eye Detector
Training Accuracy	100%	X%	X%
Training Validation Accuracy	100%	X%	X%
Test Accuracy	80%	X%	X%
Model Size	X MB	X MB	X MB

Conclusion

- What did we learn
  - Adding more convolutional layers did not always yield higher validation accuracy
  - Using
- How could be model be further improved:
  - Eye model focus only on one eye, possible false negative in few cases

Appendix

Citations

1. Nasri, I., Karrouchi, M., Snoussi, H., Kassmi, K., Messaoudi, A. (2022). Detection and Prediction of Driver Drowsiness for the Prevention of Road Accidents Using Deep Neural Networks Techniques. In: Bennani, S., Lakhrissi, Y., Khaissidi, G., Mansouri, A., Khamlichi, Y. (eds) WITS 2020. Lecture Notes in Electrical Engineering, vol 745. Springer, Singapore. https://doi.org/10.1007/978-981-33-6893-4_6
2. Ghoddoosian, R. Galib. M, Athitsos, V. (2019). A Realistic Dataset and Baseline Temporal Model for Early Drowsiiness Detection. https://openaccess.thecvf.com/content_CVPRW_2019/papers/AMFG/Ghoddoosian_A_Realistic_Dataset_and_Baseline_Temporal_Model_for_Early_Drowsiness_CVPRW_2019_paper.pdf
3. Karolinska Sleepiness Scale (KSS), https://www.med.upenn.edu/cbti/assets/user-content/documents/Karolinska%20Sleepiness%20Scale%20(KSS)%20Chapter.pdf
4. Viola, P., Jones. M., (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf
5. *wp2186170*. (n.d.). Wallpapercave. https://wallpapercave.com/wp/wp2186170.jpg
6. *How to use stratified random sampling in 2023*. (n.d.). Qualtrics. https://www.qualtrics.com/experience-management/research/stratified-random-sampling/

NeurIPS Checklist

- Do the **main claims** made in the abstract and introduction accurately reflect the paper's contributions and scope main claims?
  - yes
- Reviewed **ethics review guidelines**?:
  - yes
- Did you discuss any potential **negative societal impacts** of your work?
  - yes
- Did you describe the **limitations** of your work?
  - Yes, images not available in low lighting conditions, various other image types, rotations, other population, check for demographics. male/female, race, check possible balance
- Did you state the full set of **assumptions** of all theoretical results?
  - Yes, we assume that model works in night conditions, could be done with image process, augmentation, image provided to use comes we assumption is driver equivalent. However, no steer wheel, in car/vehicle conditions not present.Other road background conditions not present. Simulate conditions presently with students.
- Did you include complete **proofs** of all theoretical results?
  - Yes
- Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
  - Yes
- Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
  - Yes, our note notebook printed out the data splits, hyperparameters and how we chose the best model
- Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
  - Yes
- Did you include the amount of **compute** and the type of **resources** used (e.g., type of GPUs, internal cluster, or cloud provider)?
  - Yes
- If your work uses existing assets, did you cite the creators?
  - Yes

Source: NeurIPS Checklist

NeurIPS Checklist (cont.)

- Did you mention the license of the assets?
  - Yes
- Did you include any new assets either in the supplemental material or as a URL?
  - Yes
- Did you discuss whether and how consent was obtained from people whose data you're using/curating?
  - Yes
- Did you discuss whether the data you are using/curating contains **personally identifiable information** or **offensive content**?
  - Yes, this is not a problem since we sourced our data from Kaggle which is public domain.
- Did you include the full text of instructions given to participants and screenshots, if applicable?
  - N/A
- Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?
  - N/A
- Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?
  - N/A

Source: NeurIPS Checklist

Limitations

- Night time data not significantly present, simulated with low lighting conditions

- Future applications

- Current eye model limited to one eye. For future build, would consider aggregate both eyes for greater alertness level if we go beyond binary classification. How do we distinguish between just blinking and complete eyes closed due to drowsy as we only have one eye.

- Subjects not in front of actual vehicle, simulated photo images of person in chairs

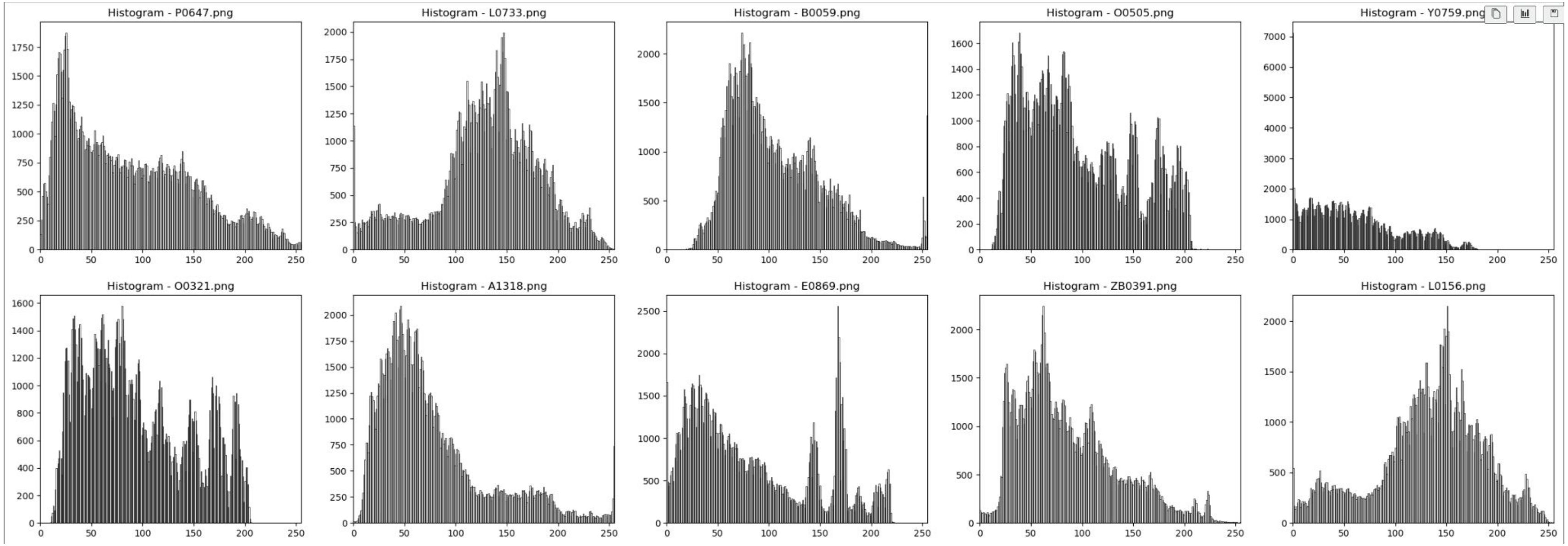
- Subjects not representative of handicap persons. People with one eye, vision issues, disabilities but able to drive.

- Subjects with sunglasses (tinted) not covered

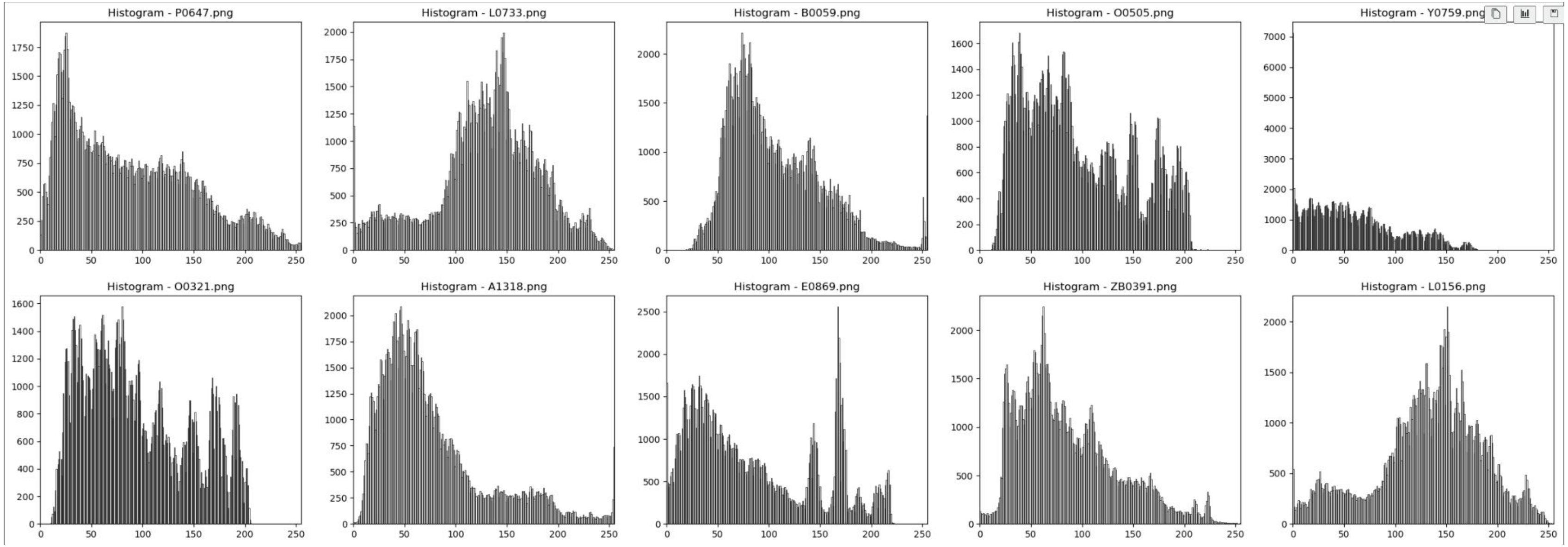
Contributions

Additional utilities and Additional Models experimented

Histogram of Drowsy class samples

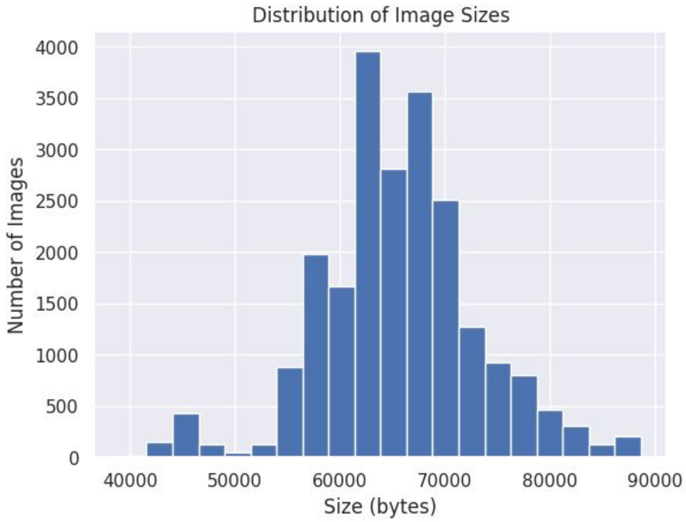


Histogram of Non Drowsy class samples

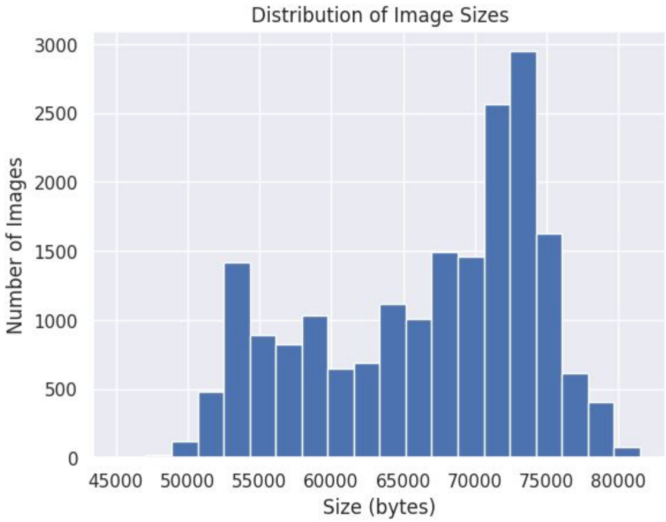


Drowsy and Non drowsy image distribution

Drowsy folder (22348 images)



Non Drowsy folder (19445 images)



RGB images, More than 41,790 images in total

Pixel Min/Min value of each class:

Drowsy:

	Image Name	Min value	Max value
0	M0560.png	0.0	255.0
1	N0215.png	0.0	233.0
2	A1346.png	0.0	255.0
3	P0343.png	0.0	255.0
4	J0199.png	0.0	255.0
5	P0525.png	0.0	255.0
6	ZB0849.png	0.0	255.0
7	L0104.png	0.0	255.0
8	ZA1061.png	0.0	255.0
9	T0903.png	0.0	255.0

Non Drowsy:

	Image Name	Min value	Max value
0	k0179.png	11.0	255.0
1	u0299.png	0.0	255.0
2	zb0791.png	3.0	255.0
3	q0320.png	3.0	255.0
4	zc0575.png	0.0	255.0
5	b0035.png	13.0	255.0
6	n0148.png	0.0	255.0
7	zc0437.png	0.0	255.0
8	s0068.png	0.0	255.0
9	a1003.png	0.0	255.0

Inter-annotator agreement (IAA)

Inter-annotator agreement (IAA) is the degree of consensus or similarity among the annotations made by different annotators on the same data. It is a measure of how well the annotators follow the same guidelines, criteria, and standards for labeling the data. Inter-annotator reliability (IAR) is the extent to which the annotations made by different annotators are valid, accurate, and trustworthy.

How do you calculate inter-annotator agreement and reliability?

There are various methods and metrics for calculating IAA and IAR, depending on the task's type, level and complexity, as well as the number and distribution of the annotators and categories. The simplest and most intuitive is percentage agreement, which calculates the proportion of data items that are identically annotated by all or a pair of annotators. Cohen's kappa adjusts this for chance agreement, producing a value between -1 and 1. Fleiss' kappa is a generalization of Cohen's kappa that can handle more than two annotators and categories, with a similar value range and interpretation.

Data Processing

- The image dataset preprocessing was executed through a two-stage process.
    - **Stage 1**:, 150 random images each were selected from directories labeled 'Drowsy' and *Non-Drowsy*' to correct for dataset imbalances and then annotated with a label of 0 (NonDrowsy) or 1 (Drowsy), facilitating binary classification.
    - **Stage 2**: dimensions of the images were standardized to a uniform size of 80x80 pixels.
- The data was then split into training, validation, and test sets in this step.
- Additionally, this stage incorporated various image augmentation techniques, including adjustments of brightness and contrast, as well as horizontal flipping of images.
- The augmented images were then systematically incorporated into the training dataset, enriching it and potentially enhancing the robustness of the model by exposing it to a more diverse range of data variations.

Final both ff and eye v3 models

Show change from model v1 to v3 changes

Hyperparameters:

learning_rate = [0.001]

optimizer = ['adam']

filters = [[16,16], [16,8], [8,16], [8,8]]

kernel_size = [[(5,5),(5,5)], [(3,3),(3,3)]]

strides = [[(1,1),(1,1)]]

pool_size = [[(2,2),(2,2)]]

conv_dropout_rate = [[0.3, 0.5]]

fc_dropout_rate = [[0.5, 0, 0]]

conv_batch_normalization = [[True, True]]

fc_batch_normalization = [[False, True, True]]

activations = [['relu', 'relu', 'relu', 'relu', 'relu']

units = [[128, 64, 32], [64, 64, 32]]

epochs = [30, 40]

batch_size = [16, 32, 64]