

Performance Analysis of Deep CNN Architectures for Facial Fatigue Detection: Superiority of ResNet50 over Benchmarks

Timothy Christyan¹ and Ade Irawan²

Department of Computer Science, Universitas Pertamina^{1,2}

Center for Data Science and Automation (CDSCAN), Universitas Pertamina²

tchristyan123@gmail.com¹, adeirawan@universitaspertamina.ac.id²

Article Info

Article history:

Received Oct 31, 2025

Revised Nov 25, 2025

Accepted Aug 31, 2026

Keyword:

Facial Fatigue

Algorithm

CNN

ResNet

Model

Accuracy

ABSTRACT

Increased screen time exposure to digital screens can cause fatigue and harm health. The facial fatigue detection system is one solution for monitoring the condition of digital screen users. The latest research in this field has proposed a facial fatigue detection system using a combination of the FaceNet algorithm with k-Nearest Neighbor (kNN) or multiclass Support Vector Machine (SVM), with an accuracy of 94.68%. This paper studies the use of Convolutional Neural Network (CNN) with three architectures, namely AlexNet, Visual Geometry Group (VGG), and Residual Neural Network (ResNet), to increase the accuracy of the previous research. The UTA-RLDD facial dataset, which consists of three classes, i.e., tired, alert, and non-vigilant, is used to validate the performances. Facial features are taken from the dataset using the Haar Cascades algorithm. After preprocessing and face extraction using the Haar Cascades algorithm, a total of 3446 facial images are used for model training and evaluation. The facial features are then resized and used to train the models; splitted into around 2400 images for training, 300 images for validation, and another 300 for testing. It is found that the ResNet architecture with 50 layers (ResNet50) delivers higher accuracy on the testing set compared to other architectures, which is 95.93%. The results also show that adding architectural parameters does not always increase the accuracy.

© This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Corresponding Author:

Ade Irawan

Department of Computer Science

Universitas Pertamina

Jl. Teuku Nyak Arief, Simprug, Jakarta, Indonesia

adeirawan@universitaspertamina.ac.id

1. INTRODUCTION

Advances in information technology have enabled widespread reliance on screen-based devices to support work-related activities. Furthermore, the COVID-19 pandemic significantly increased screen usage intensity. This pattern of behavior has continued beyond the pandemic period, as surveys and research conducted by the Pew Research Center during the pandemic indicate that 78% of American workers prefer to continue working online [1]. As a result, increased exposure to digital device screens, such as computers and smartphones, became inevitable. The convenience and benefits offered by online working systems—such as flexible working hours, eliminating commutes, and reduced company expenses—have led to their continued implementation even after the pandemic ended.

Increasing screen time exposure to digital devices can have harmful effects on the body. Research by [2] and [3] concludes that prolonged exposure to screens on digital devices can cause sighting problems and hinder brain growth. Research by [4] says that screen time exposure should be limited to less than two hours a day. One problem that can occur due to exposure to long screen time is Computer Vision Syndrome (CVS) [5]. CVS symptoms are difficult to recognize, especially when busy with activities in front of a digital device screen. A facial fatigue detection system can address this issue by monitoring fatigue caused by prolonged exposure to screens and can be directly implemented on digital devices for real-time detection [6].

The facial fatigue detection system utilizes developments in Artificial Intelligence and Computer Vision, making it possible to create image classification models that are able to learn more features in an image. The Convolution Neural Network (CNN) [7] and its derivatives are part of the Artificial Intelligence algorithm that is capable of performing various facial image classification tasks, such as detecting sleepy faces in motor vehicle drivers [8] and facial mask detection [9]. However, detecting facial fatigue as a multi-class problem remains challenging due to the high similarity between fatigue levels, gradual transitions between classes, and significant inter-subject variability in facial expressions. Fatigue-related facial cues can also be influenced by external factors such as lighting conditions, camera angles, facial anatomy, and individual behavioral differences, making class boundaries less distinct. These factors increase the difficulty of accurately distinguishing multiple fatigue states, thereby motivating further investigation into robust CNN architectures and feature extraction strategies for multi-class facial fatigue detection.

Previous research [6] utilized a CNN derivative, namely the FaceNet architecture combined with kNN and SVM classifiers, to design a facial fatigue detection system caused by prolonged screen exposure. The proposed model achieved a test precision of 94.68%. However, this study relied on a single CNN-based feature extractor and did not analyze the impact of CNN architectural variations and network depth on performance. Although previous approaches used feature embeddings with shallow classifiers, this work explores end-to-end feature learning to improve accuracy. Therefore, this study addresses the following research questions: (1) how different CNN architectures affect facial fatigue detection performance, (2) whether network depth influences detection accuracy, and (3) which architecture achieves the best overall performance.

The main contributions and novelty of this paper are as follows. First, unlike previous studies that relied on a single CNN-based feature extractor, this work presents a systematic comparative analysis of multiple CNN architectures—namely AlexNet [10], VGG [11], and ResNet [12]—for the detection of facial fatigue in the same experimental settings. Second, we explicitly investigate the role of network depth in fatigue detection performance and demonstrate that deeper residual networks significantly improve accuracy. In particular, ResNet-50 achieves the best performance, with an accuracy of 95.93%, surpassing previously reported results. Finally, this study provides new

experimental evidence on how architectural depth and residual learning affect facial fatigue detection, offering practical guidance for model selection in future fatigue-related computer vision applications.

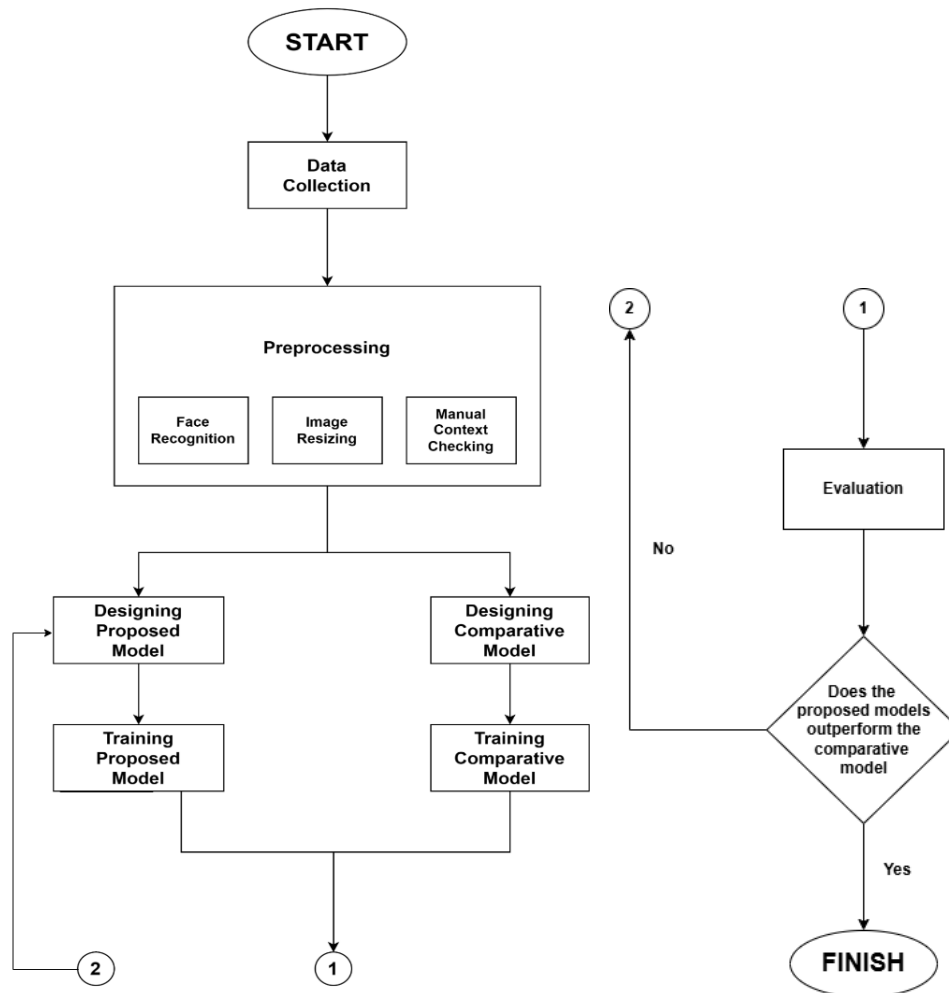


Figure 1. The research method.

Classical CNN architectures such as AlexNet [10] and VGG [11] rely on sequential convolutional and pooling layers without explicit mechanisms to preserve gradient flow across depth. As a result, these models are more susceptible to optimization issues, including vanishing gradients and inefficient feature reuse, particularly when applied to fine-grained visual tasks such as facial fatigue detection, where subtle texture and expression cues must be captured. In contrast, deep residual networks such as ResNet [12] introduce skip connections that facilitate stable gradient propagation and enable deeper learning using more discriminative representations. This architectural difference allows residual networks to better model subtle fatigue-related facial patterns while maintaining training stability, thereby explaining their superior performance compared to classical CNNs in this application.

This paper is organized as follows. Section 1 discusses the background of this research. Section 2 discusses research methods, including data collection, data preprocessing, and model training.

Section 3 discusses the results of the training process and compares them with previous research. Section 4 provides the conclusions of this paper.

2. RESEARCH METHOD

This research was conducted in several stages, as shown in Figure 1. It started with data collection, followed by data preprocessing in the form of facial feature extraction from the data set, and then designing models and training using processed data. In the end, we evaluated the models, where the metrics of each model were compared not only with each other, but also with the baseline model from previous research.

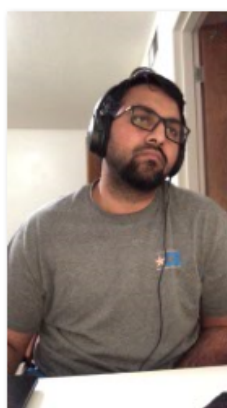
2.1. Dataset

This research utilized facial image data of people using a video conferencing application in three facial states: focused, nonvigilant, and tired, where each sample is shown in Figure 2. People's face data were taken from the UTA-RLDD dataset [13]. The data set contains facial data from 60 volunteers taken using a video conferencing application for 30 hours. The data set was compiled to design a sleepy face detection model. However, based on previous research, it can also be used to detect early signs of tired faces [6].

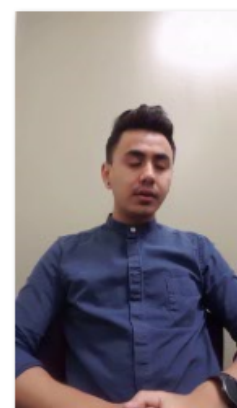
As the dataset is derived from video recordings, extracted frames may exhibit temporal correlation [14], particularly when consecutive images are captured under similar background and facial conditions, which could potentially introduce data leakage if not handled carefully. In this study, the data splitting strategy follows the same method as previous work [6] to ensure a fair and consistent comparison of results. Consequently, any potential temporal dependency present in the dataset affects all evaluated models equally and does not bias the comparative analysis. The dataset is divided into training, validation, and testing subsets for model development and evaluation.



Focus



Non Vigilant



Tired

Figure 2. Example of face image [13].

2.2. Data Preprocessing

The preprocessing stage consisted of two steps: face detection and image resizing. Prior to preprocessing, the dataset was randomly shuffled to eliminate any ordering bias and to ensure a balanced distribution of samples during subsequent training and evaluation stages. The face detection process is carried out using the Haar Cascades facial object detection algorithm [15]. In this preprocessing step face regions were detected using the Haar Cascade frontal face detector provided by OpenCV (*haarcascade_frontalface_default.xml*). Each input image was first converted from RGB to grayscale. The `detectMultiScale` function was applied with a scale factor of 1.3 and `minNeighbors` of 5, which control the image pyramid scaling and the strictness of face candidate validation. The Detected face bounding boxes were then used to crop the facial region from the original image.

For image resizing, facial images resulting from the object detection stage are converted into dimensions (227×227 and 224×224). For example, the data preprocessing results from the samples in Figure 2 are shown in Figure 3. Aside from face detection and image resizing, no additional image normalization was applied during preprocessing. The cropped face images were resized to match the input resolution required by the model and represented as three-channel NumPy arrays. The original BGR color channel order, as provided by OpenCV, was preserved, and pixel intensity values were maintained in their original 8-bit integer range (0–255). No pixel scaling to [0,1], mean–standard deviation normalization, or color space transformation was performed in preprocessing step.

Data preprocessing outcomes could still experience errors, where the facial object detection process could falsely predict other object as face (e.g., wearing clothes that have a face in it, other object or background that looks or have similar feature to human face, etc). In this case, we manually checked and delete the incorrect data. The amount of data in each part of the dataset, after and before preprocessing, is shown in Table 1. Approximately 19.5% of the images are discarded during preprocessing. Specifically, around 15.4%, 15.8%, and 26.2% from the alert, non-vigilant, and tired classes, respectively.

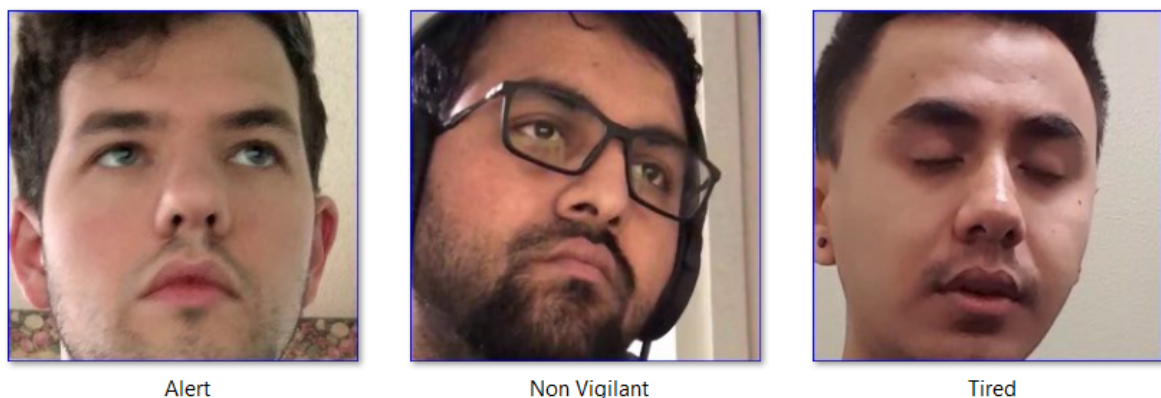


Figure 3. Example of face image after preprocessing.

Table 1. Class distribution of the dataset before and after preprocessing.

	Raw Data			Preprocessed Data		
	Train	Validation	Test	Train	Validation	Test
Alert	1001	125	126	841	108	110
Non Vigilant	975	121	123	818	100	109
Tired	1148	143	145	838	121	101

2.3. Classification using CNN

This research uses three types of CNN derivatives: AlexNet, Visual Geometry Group (VGG), and Residual Neural Network (ResNet). The following subsections provide brief explanations of those.

2.3.1. AlexNet

The AlexNet architecture consisted of eight layers, i.e., five convolution layers and three fully connected layers, as shown in Figure 4. Furthermore, it implements several parameter optimization methods, such as dropout [16] to overcome overfitting, Local Response Normalization [17] to limit the output range, and Rectified Linear Units (ReLU) [18] to be an activation function. The AlexNet architecture succeeded in improving CNN's learning capabilities when it was proposed by increasing depth, i.e., using seven stages of feature extraction, whereas the state-of-the-art architecture at that time only used five stages.

Although AlexNet represented a significant advancement at the time of its introduction by increasing network depth compared to earlier CNNs, its relatively shallow architecture limits its ability to capture fine-grained facial details and subtle micro-expressions that are characteristic of fatigue, such as slight eyelid drooping, weak muscle tension, and minor texture changes around the eyes and mouth. As a result, AlexNet was expected to rely primarily on coarse spatial features, which may reduce its sensitivity to low-intensity fatigue signals compared to deeper architectures.

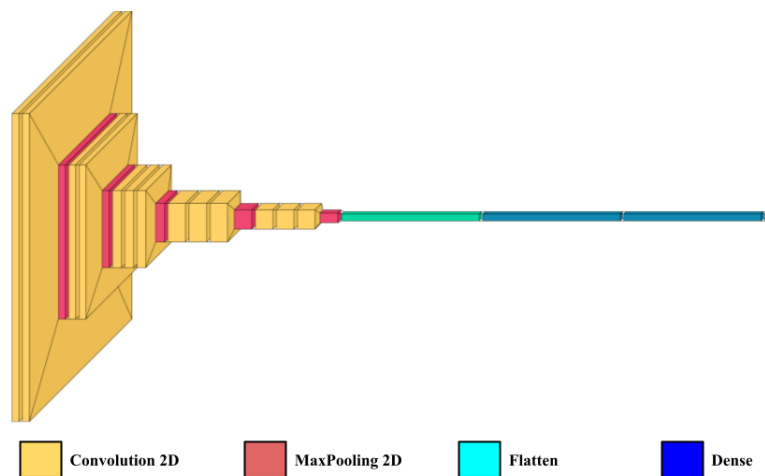


Figure 4. AlexNet architecture.

2.3.2. Visual Geometry Group (VGG)

The VGG architecture consists of eight to sixteen convolution layers, some pooling layers, and three fully connected layers, as shown in Figure 5. The architecture was developed to determine the influence of the architectural depth on the precision of the prediction. The filter used in each convolution layer in the VGG architecture has a constant size of 3×3 to reduce the influence on the training process, so the performance is more influenced by the depth of the model.

In the context of facial fatigue detection, the increased depth of VGG enables the extraction of more detailed features, which is beneficial to capture subtle facial patterns associated with fatigue, such as gradual eye closure, changes in facial texture, and weak muscle activation. However, due to the absence of residual connections, deeper VGG variants are more susceptible to vanishing gradient effects, which can hinder effective learning of low-intensity fatigue cues and limit performance gains as network depth increases.

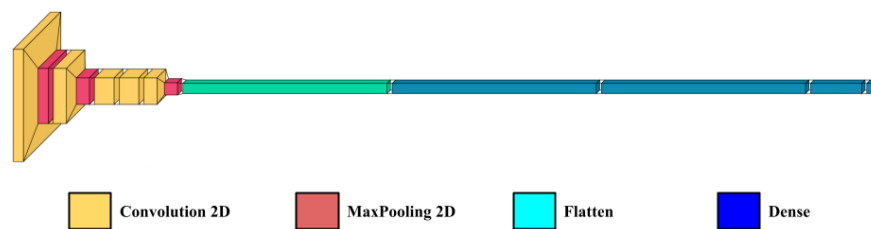


Figure 5. VGG16 architecture.

2.3.3. Residual Neural Network (ResNet)

The ResNet architecture was designed to overcome the vanishing gradient problem [19], which caused the limited depth for CNN architectures when it was proposed. ResNet does this by applying "skip connections" technique, which skips several convolution layers and uses the results of the activation function from the previous layer, as illustrated in Figure 6. Skip connections are generally performed every two or three convolution layers.

The general ResNet architecture consists of 18 to 152 layers consisting of one fully connected layer and as many convolution layers as the remaining layers require. By preserving gradient flow during backpropagation, ResNet enables the training of significantly deeper networks. In the context of facial fatigue detection, this architectural design facilitates more effective learning of subtle and low-intensity facial features, such as slight eyelid drooping, reduced muscle tension, and fine texture variations, which are often lost in shallower or non-residual networks.

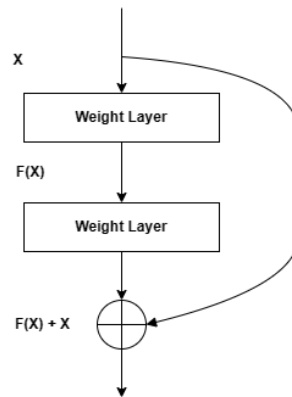


Figure 6. ResNet skip connection.

3. RESULT AND ANALYSIS

This research is carried out in the Google Collaboratory environment. Three models, which are the proposed models in this paper, are trained using AlexNet, VGG, and ResNet. On the other hand, one model is trained using the FaceNet algorithm with k-Nearest Neighbor (kNN) as baseline.

For all models, performance analysis was based on training and validation accuracy trends, loss trends, precision, recall, f1-score, and confusion matrix obtained from the training process. All networks were optimized using the Adam optimizer [20] with its default learning rate of 0.001. No early stopping or learning rate scheduling strategies were applied during training.

3.1. Model Training Using AlexNet

The AlexNet model was trained using categorical cross-entropy it's loss function. No explicit regularization techniques, such as dropout or L2 weight decay, were applied in the network architecture; instead, training stability was supported through the use of ReLU activations and He-normal weight initialization. Training was performed using a fixed default batch size for all experiments, and the model was optimized end-to-end until convergence. The model was trained for 100 epochs, resulting in a total training time of approximately 7.5 minutes.

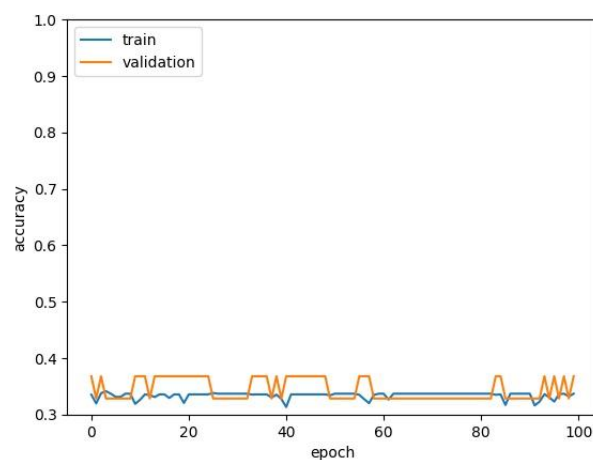


Figure 7. Training using AlexNet.

Table 2. AlexNet Class Performance on Testing Data.

	Precision	Recall	F1-score
Alert	0.00	0.00	0.00
Non Vigilant	0.00	0.00	0.00
Tired	0.32	1.00	0.48

The input received by the architecture is image data in RGB format with size 227×227 . The graph of training performance after 100 epochs is shown in Figure 7. The figure shows that the validation accuracy does not increase higher than 36%, while the training accuracy remains constant around 30%. This was further supported by low precision (32%) and F1-score (48%) which can be seen in Table-2. The low accuracy can be resulted from architecture failure to learn certain facial features in the dataset, caused by the model only classifying all image data into one class.

Other reason for AlexNet poor performance is vanishing gradient issues. This is supported by stagnant training and validation accuracy, and also by the corresponding loss curves. As shown in the loss curve Figure 8, the training loss decreases sharply during the initial epoch and then remains nearly constant, while the validation loss exhibits minimal change across subsequent epochs. This behavior indicates that the model quickly converges to a suboptimal solution and fails to further minimize the loss, suggesting optimization difficulties rather than overfitting.

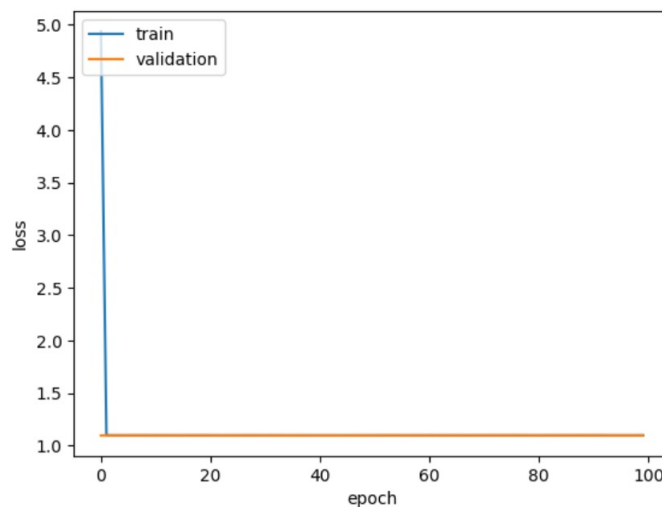


Figure 8. AlexNet Loss Graph.

Further evidence is provided by the confusion matrix in Figure 9 and the recall score in Table 2 that show a perfect score in one class but zero for other, revealing that AlexNet collapses into predicting a single dominant class, demonstrating an inability to learn discriminative facial features across categories. The absence of adaptive learning rate control may have further limited the model's ability to escape poor local minimal.

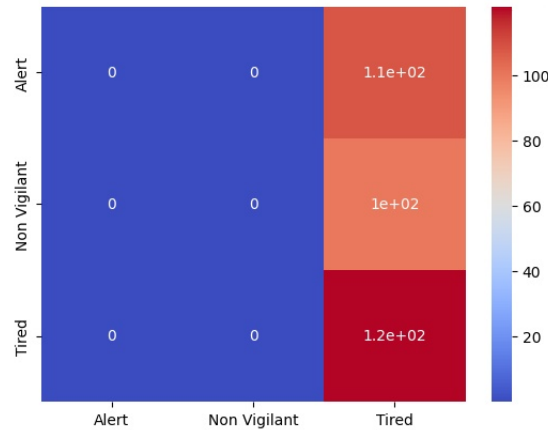


Figure 9. Testing Confusion Matrix for AlexNet model.

3.2. Model Training Using VGG

Two VGG architectures are evaluated in this paper: a sixteen-layer VGG (VGG16) consisting of thirteen convolution layers and three fully connected layers, and a nineteen-layer VGG (VGG19) with sixteen convolution layers and three fully connected layers. The architecture receives image data in RGB format with size 224×224 as input. The VGG16 and VGG19 models were implemented using the Keras application framework and trained from scratch. The weights were initialized randomly (*weights = None*) and the full classification head included (*include_top = True*). Both models accepted input images of size $224 \times 224 \times 3$, consistent with the original VGG design.

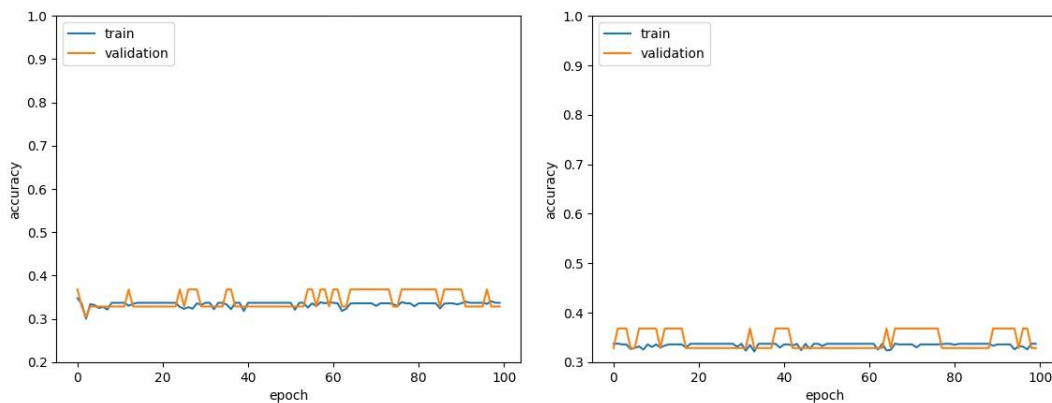


Figure 10. Training using VGG architecture (right = VGG16, left = VGG19)

Training was performed using the same optimization settings as AlexNet to ensure a fair architectural comparison. No explicit regularization techniques, such as dropout or L2 weight decay, were applied beyond the inherent architectural design of the VGG networks. Each model was trained for 100 epochs, and the total training time was around 55.5 minutes for VGG16 and around 1.5 hours for VGG19. Training performance is shown in Figure 10.

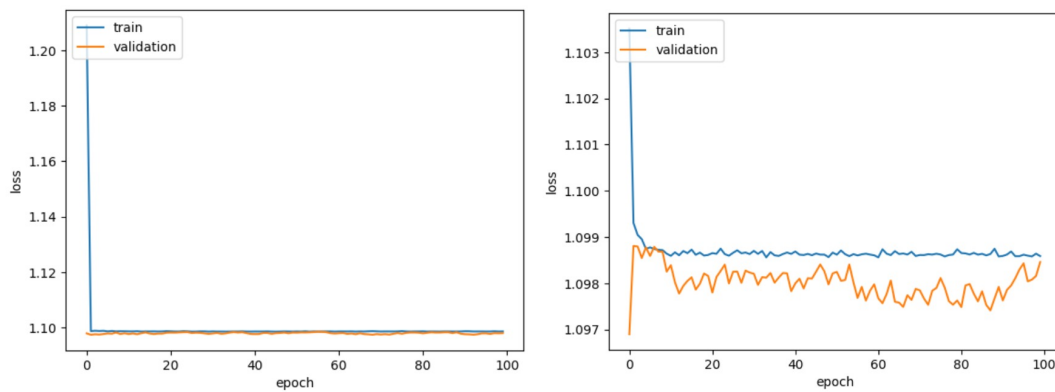


Figure 11. VGG Loss Graph (right = VGG16, left = VGG19).

Training using the VGG architecture shows results that are not very different from those of AlexNet. The model accuracy did not increase higher than 36% and remained constant between 30%. Hence, the model trained with VGG might experience the same issue as the AlexNet model.

Table 3. VGG Class Performance on Testing Data.

	VGG16			VGG19		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Alert	0.00	0.00	0.00	0.00	0.00	0.00
Non Vigilant	0.00	0.00	0.00	0.00	0.00	0.00
Tired	0.32	1.00	0.48	0.32	1.00	0.48

Similarly to the AlexNet results, the limited performance of the VGG16 and VGG19 models is supported by stagnant training and validation accuracy, and also by the corresponding loss curves Figure 11. Both models exhibit a sharp decrease in training loss during the initial epochs, followed by an extended flat line where training and validation losses remain nearly constant across subsequent epochs. The similarity between training and validation loss indicates that the models do not suffer from overfitting; instead, they stuck at suboptimal solutions, suggesting optimization difficulties during training.

Further evidence is provided by the confusion matrix analysis in Figure 12 and recall value of one in one class but zero for other as shown in Table 3. This shows that both VGG architectures predominantly predict a single class, reflecting an inability to learn facial features across target categories. The absence of adaptive learning rate control likely limited effective gradient propagation in deeper VGG networks trained from scratch. These results indicate that the failure of VGG models is primarily related to architectural and training limitations rather than dataset imbalance.

3.3. Model Training Using ResNet

This research evaluated three ResNet architectures: ResNet with 50 layers, 101 layers, and 152 layers. Each architecture divides the convolution layers into residual blocks with three convolution layers for each block. The parameters for the number of residual blocks and the number of convolution

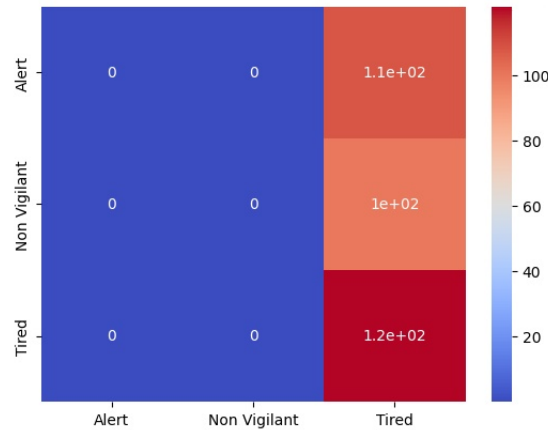


Figure 12. Testing Confusion Matrix for VGG16 and VGG19 model (Both give the same result).

layers for each ResNet architecture used are shown in Table 4. The skip connection process in the ResNet architecture is carried out between one residual block and the residual block after it.

The ResNet models were implemented using Keras ResNet application framework and trained from scratch, with randomly initialized weights (*weights = None*) and the full classification head included (*include_top = True*). All ResNet variants accepted input images of size $224 \times 224 \times 3$, consistent with the original ResNet design. Training was conducted using the same optimization settings as AlexNet and VGG to ensure a fair comparison across architectures. No explicit regularization techniques, such as dropout or L2 weight decay, were applied beyond the inherent residual connections of the ResNet architecture. Each model was trained for 100 epochs, and the total training time was measured around 59.15 minutes for ResNet50, around 1.6 hours for ResNet101, and 2.1 hours for ResNet152.

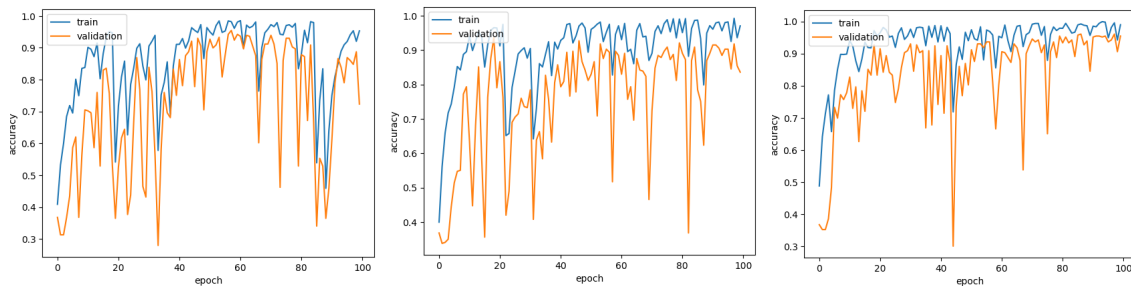


Figure 13. Training using ResNet architecture (right = ResNet50, middle = ResNet101, left = ResNet152).

Figure 13 shows the training performances of those models. ResNet50 provides higher accuracy than the other two ResNet models, 96.05%. On the other hand, ResNet101 gives the best accuracy of 95.44% and ResNet152 of 93.61%. The training plot also shows that ResNet50 architecture have better learning ability than ResNet101 and ResNet152 because it is more stable overall. The smaller number of trainable parameters in ResNet50 may be the cause. The increasing number of trainable parameters in ResNet101 and ResNet152 might also cause the model to learn

Table 4. ResNet Architecture convolution layer and residual block.

	Convolution Layer	Residual Block
ResNet50	41	16
ResNet101	99	33
ResNet152	150	50

parameters that have little to do with the feature or even interfere with the training process. This can be seen by how the ResNet50 model gives the highest accuracy using the less trainable parameters among the three ResNet models.

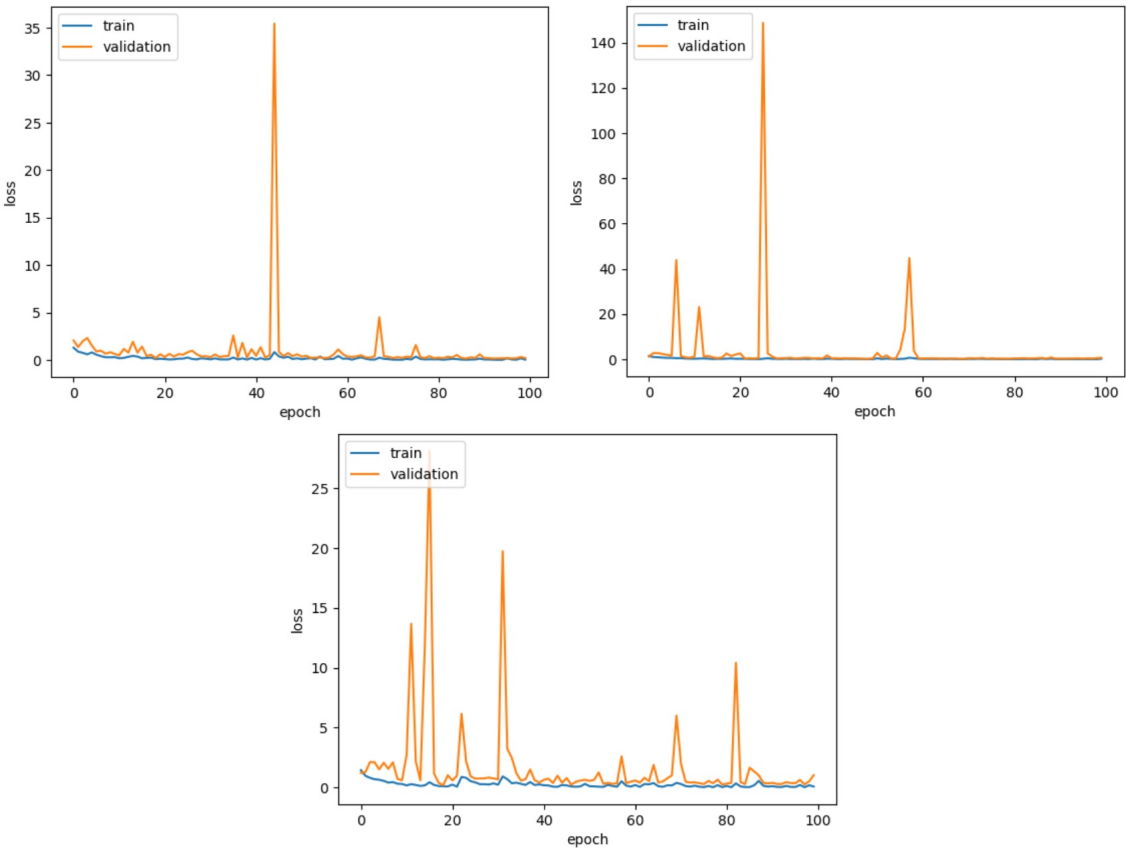


Figure 14. ResNet Loss Graph (right = ResNet50, middle = ResNet101, left = ResNet152).

In contrast to AlexNet and VGG architectures, the ResNet models demonstrate improved learning behavior, which can be attributed to the use of residual connections that facilitate gradient propagation. This observation is supported by both accuracy trends in Figure 13 and the corresponding training and validation loss curves in Figure 14. As shown in the loss plots, all ResNet variants achieve a rapid reduction in training loss. Unlike AlexNet and VGG, where a similarly rapid loss decrease is accompanied by stagnant training accuracy and collapse into single-class predictions, the ResNet models achieve a concurrent and substantial increase in training accuracy, exceeding 88%.

This distinction suggests that the loss reduction in ResNet reflects meaningful feature learning rather than degenerate optimization.

Table 5. ResNet Model Class Performance on Testing Data.

	ResNet50		
	Precision	Recall	F1-score
Alert	0.97	0.95	0.96
Non Vigilant	0.97	0.96	0.97
Tired	0.93	0.96	0.95

	ResNet101			ResNet152		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Alert	0.98	0.90	0.94	0.93	0.96	0.95
Non Vigilant	0.91	0.88	0.89	0.92	0.90	0.91
Tired	0.86	0.96	0.91	0.93	0.91	0.92

Further evidence is provided by the confusion matrix analysis shown in Figure ??, where all ResNet variants display more balanced class predictions. Table 5 also show that overall, ResNet have a far better testing precision, recall, and f1-score metric than AlexNet and VGG, which collapsed into a single dominant class. Among the ResNet models, ResNet50 demonstrates the clearest class separation and the fewest misclassifications. The absence of adaptive training strategies may have disproportionately affected deeper ResNet variants, as their larger number of trainable parameters increases sensitivity to optimization instability.

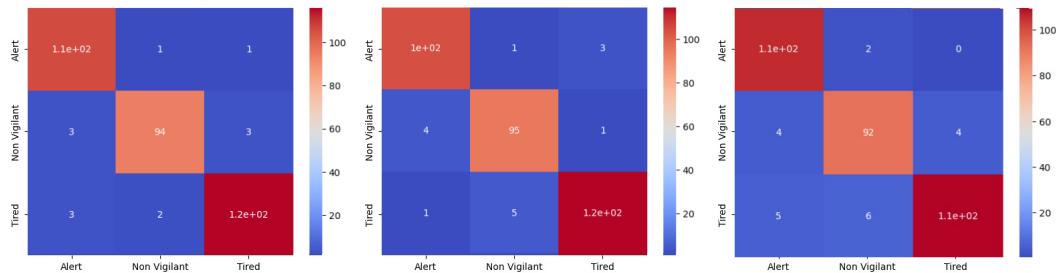


Figure 15. Testing Confusion Matrix for ResNet Models (right = ResNet50, middle = ResNet101, left = ResNet152).

Overall, these results indicate that residual learning effectively mitigates the optimization failures observed in AlexNet and VGG, while also highlighting that moderate depth, as in ResNet50, provides a more favorable balance between representational capacity and training stability.

3.4. Model Training Using FaceNet + kNN

Training using the FaceNet facial feature extraction algorithm and kNN classification algorithm [6] was used as baseline comparison. The parameters used include FaceNet with output in the form

of an embedding with 128 dimensions and kNN with a value of $k = 1$. Training results are a model with 22,808,144 trainable parameters and an accuracy of 96.96%.

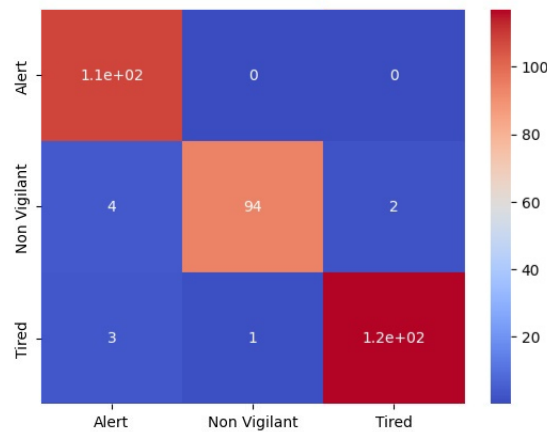


Figure 16. Testing Confusion Matrix for FaceNet + kNN model.

3.5. Analysis

The training results in Table 6 show that the AlexNet and VGG models failed to learn from the data, as shown by their low training and testing accuracies. This likely caused by optimization problem, such as the vanishing gradient, which commonly affects classical CNN architectures. In contrast, residual learning enables deeper networks to converge more effectively by preserving gradient flow across layers. Can be seen in ResNet model with 50 layers (ResNet50) which have the highest training accuracy of 96.04%, outperforming both shallower CNNs and deeper residual variants (ResNet101 and ResNet152).

Meanwhile, the comparison baseline method achieves the highest training accuracy of 96.96%. The training performance plots indicate that none of the evaluated models continue to learn new discriminative features toward the end of the training process. This behavior is reflected by the stabilization of the accuracy curves, which become relatively linear and exhibit minimal variation in the final epochs. For loss curve, every proposed model able to fastly optimize in the early epochs, even though both AlexNet and VGG architecture is stuck at one class. This was also supported by the confusion matrix and other class performance (e.g., precision, recall, and f1-score).

The simulation results show that ResNet50 has the highest testing accuracy of 95.93%. On the other hand, the baseline model produced an accuracy of 94.06%. The ResNet50 model also has the smallest number of trainable parameters from all the proposed models, i.e., 23,540,739, which is only slightly higher than the comparison model which is 22,779,312. Although deeper networks have higher representational capacity, their performance may degrade when trained on limited dataset due to overfitting, inefficient parameter utilization, or vanishing gradient problem. The UTA-RLDD dataset [13] may not be sufficient to fully exploit the additional depth of ResNet101 and ResNet152, leading to reduced generalization performance compared to ResNet50. This suggests that ResNet50 provides an optimal balance between depth and learnability for the given task and dataset.

The experimental results also indicate that increasing the number of trainable parameters does not necessarily lead to higher accuracy. This can be seen from how the ResNet50 model has

Table 6. Training Results

	Trainable Parameter	Training Accuracy	Testing Accuracy
AlexNet	62, 381, 347	36.77%	31.56%
VGG16	134, 272, 835	36.77%	31.56%
VGG19	139, 582, 531	36.77%	31.56%
ResNet50	23, 540, 739	96.04%	95.93%
ResNet101	42, 558, 979	95.44%	92.5%
ResNet152	58, 225, 667	93.61%	92.5%
FaceNet + kNN	22, 779, 312	96.96%	94.06%

higher accuracy than other models, even though having fewer parameters than deeper variants such as ResNet101 and ResNet152, as well as parameter-heavy architectures like AlexNet and VGG. This highlights that architectural efficiency and effective feature reuse enabled by residual connections are more critical than parameter count alone in facial fatigue detection.

Furthermore, architectures with a very large number of parameters, such as AlexNet, VGG16, and VGG19, exhibit poor learning performance in this task. This behavior may be attributed to optimization difficulties, including the vanishing gradient problem, as well as potential over-parameterization, where additional parameters contribute little useful information and may even hinder effective feature learning. These results demonstrate that there is no linear relationship between the number of parameters and model accuracy; instead, an optimal balance between model complexity and architectural efficiency is required for facial fatigue detection.

From a computational perspective, ResNet50 also have advantages in terms of model size and inference. Compared to ResNet101 and ResNet152, ResNet50 needs fewer operations and lower memory consumption, resulting in faster inference times. These properties make ResNet50 more suitable for real-time facial fatigue monitoring systems, where timely predictions and limited computational resources are critical constraints. In practical deployment scenarios such as fatigue monitoring through laptop webcams, real-time surveillance during prolonged digital device usage, or implementation on low resource mobile devices, the trade off between accuracy and computational cost is essential. The results of this study indicate that ResNet50 achieves high detection accuracy while still maintaining relatively low model complexity, making it a strong candidate for real-world fatigue detection applications.

4. CONCLUSION

This research investigates facial fatigue detection using CNN models based on deep learning in three representative architectures: AlexNet, VGG, and ResNet. The experimental results demonstrate that architectural design and network depth play a more critical role in determining detection performance than the sheer number of parameters. Among all models evaluated, ResNet50 achieves the highest test precision of 95.93%, outperforming both shallower CNNs and deeper variants, and surpassing previous work based on a FaceNet–kNN combination (94.68%).

From a practical perspective, the results indicate that increasing the complexity of the model does not always guaranty an improvement in accuracy, since overly parameterized architectures may suffer from optimization difficulties and inefficient feature learning. The superior performance of ResNet50 highlights the effectiveness of residual connections in facilitating stable training and robust feature extraction for facial fatigue detection. These findings provide practical insights for selecting CNN architectures, suggesting that moderately deep residual networks offer an optimal balance

between accuracy and model complexity for real-world fatigue detection applications.

Despite these promising outcomes, this study has several limitations. All experiments were conducted using the UTA-RLDD dataset, which is collected under specific recording conditions and contains a limited range of subjects and demographic diversity. Resulted in limitation of the generalizability of the trained models to broader populations, different illumination conditions, or unconstrained environments. The models also trained without transfer learning, explicit regularization, learning rate scheduling, or early stopping, which may have constrained the performance of deeper architectures.

Future work may address these limitations by incorporating transfer learning from large-scale facial representation models to improve feature generalization, applying data augmentation techniques to enhance robustness against variations in illumination and pose, and exploring multi-modal fatigue detection frameworks that combine facial cues with additional features such as eye-blink dynamics, head pose, or physiological measurements. These directions have the potential to further improve accuracy and reliability in practical fatigue monitoring applications.

REFERENCES

- [1] K. Parker, J. M. Horowitz, and R. Minkin, "Covid-19 pandemic continues to reshape work in america. pew research center," February 2022. [Online]. Available: <https://www.pewresearch.org/social-trends/2022/02/16/>
- [2] S. Poudel, "Research report about effect of display gadgets on eyesight quality (computer vision syndrome) of m.sc.(csit) students in tribhuvan university," *International Journal of Scientific and Engineering Research*, vol. 9, no. 8, pp. 22–32, August 2018.
- [3] E. J. Neophytou, L. A. Manwell, and R. Eikelboom, "Effects of excessive screen time on neurodevelopment, learning, memory, mental health, and neurodegeneration: a scoping review," *International Journal of Mental Health and Addiction*, vol. 19, no. 7, December 2019.
- [4] NHLBI, "Reduce screen time," February 2013. [Online]. Available: <https://www.nhlbi.nih.gov/health/educational/wecan/reduce-screen-time/index.htm>
- [5] Z. Yan, L. Hu, H. Chen, and F. Lu, "Computer vision syndrome: A widely spreading but largely unknown epidemic among computer users," *Computers in Human Behavior*, vol. 24, no. 5, pp. 2026–2042, September 2008.
- [6] F. D. Adhinata, D. P. Rakhmadani, and D. Wijayanto, "Fatigue detection on face image using facenet algorithm and k-nearest neighbor classifier," *Journal of Information Systems Engineering and Business Intelligence*, vol. 7, no. 1, pp. 22–30, April 2021.
- [7] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: <http://arxiv.org/abs/1511.08458>
- [8] I. Nagappan, J. V. T. Abraham, A. Muralidhar, and V. R. Kalluri, "Real-time driver fatigue or drowsiness detection system using face image stream," *International Journal of Civil Engineering and Technology*, vol. 8, no. 11, pp. 793–802, November 2017.
- [9] F. Amer and M. S. H. Al-Tamimi, "Face mask detection methods and techniques: A review," *The International Journal of Nonlinear Analysis and Applications (IJNAA)*, vol. 13, no. 1, pp. 3811–3823, February 2022.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, january 2012.
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

- [12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [13] R. Ghoddoosian, M. Galib, and V. Athitsos, “A realistic dataset and baseline temporal model for early drowsiness detection,” 2019.
- [14] B. Pan, R. Panda, C. Fosco, C.-C. Lin, A. Andonian, Y. Meng, K. Saenko, A. Oliva, and R. Feris, “Va-red²: Video adaptive redundancy reduction,” 2021. [Online]. Available: <https://arxiv.org/abs/2102.07887>
- [15] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” vol. 1, 02 2001, pp. I–511.
- [16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012. [Online]. Available: <https://arxiv.org/abs/1207.0580>
- [17] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-performance neural networks for visual object classification,” 2011. [Online]. Available: <https://arxiv.org/abs/1102.0183>
- [18] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for lvcsr using rectified linear units and drop-out,” *2013 IEEE international conference on acoustics, speech and signal processing*, May 2013.
- [19] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” 2013. [Online]. Available: <https://arxiv.org/abs/1211.5063>
- [20] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>