Facial Orientation Detection: Vision Transformers vs. Convolutional Neural Networks

1st Sebastian Tremblay
Northeastern University
Boston, MA
tremblay.se@northeastern.edu

1st Alexander Angione
Northeastern University
Boston, MA
angione.a@northeastern.edu

***Abstract*—This papers provides a comprehensive overview of CNN-based and ViT-based approaches to face detection. We discuss representative models, compare their performance and design philosophies, and examine how they fare on standard face detection benchmarks.**

I. Literature Review

A. Introduction

Face detection is the computer vision task of locating human faces within digital images. It is a critical first step in many applications such as face recognition, emotion analysis, and facial motion capture. Over the past two decades, face detection methods have evolved dramatically.

The rise of deep learning, and in particular Convolutional Neural Networks (CNNs), brought a paradigm shift to face detection. Starting in the mid-2010s, CNN-based detectors began to significantly outperform traditional methods by learning robust features directly from data. More recently, Vision Transformers (ViTs) have emerged as an alternative deep architecture, utilizing self-attention mechanisms to model global relationships in an image [9]. This literature review provides a comprehensive overview of CNN-based and ViT-based approaches to face detection.

B. CNN-Based Face Detection

Convolutional Neural Networks have been the dominant approach to face detection since the mid-2010s. CNN-based detectors typically operate by sliding a learned kernel over the image identify candidate face regions, followed by classification and bounding-box regression. One of the earliest successful CNN face detectors was the CNN cascade introduced by Li *et al.* in 2015 [6]. Cascading is a multi-stage detection strategy involving a sequence of classifiers (stages). The classifiers began as efficient, simple stages that quickly reject obvious negative regions, quickly filtering inputs for subsequent, more complex and accurate classifiers.

Around the same time, researchers began exploring multi-task learning to improve detection robustness. Zhang *et al.* (2016) proposed the Multi-Task Cascaded CNN (MTCNN), which jointly performs face detection and facial landmark localization (alignment) in a three-stage cascade [19]. By training small CNNs to first propose candidate face windows, then refine them, and simultaneously predict facial landmarks, MTCNN improved precision while remaining efficient for real-time use, becoming a widely used baseline [19].

Following these initial successes, a plethora of advanced CNN-based face detectors were developed, often inspired by innovations in generic object detection. In generic object detection, two major paradigms exist: two-stage detectors that generate region proposals and then classify them, and one-stage detectors that directly predict object locations in a single pass [20].

Face detection, being a single-class detection problem, particularly benefited from one-stage CNN designs that could be simplified and optimized for faces. In 2017, Zhang *et al.* introduced **S^{3}FD: Single Shot Scale-Invariant Face Detector**, which adapted the Single Shot Detector (SSD) framework to faces and explicitly addressed the challenge of detecting faces at vastly different scales [20]. S^{3}FD used a multi-scale feature pyramid (i.e. it made predictions from several convolutional layers of different resolutions) to ensure both small and large faces were adequately represented [20]. As a result, S^{3}FD achieved state-of-the-art accuracy on common benchmarks like WIDER FACE while maintaining practical speed [20].

By 2019–2020, CNN-based face detectors had matured to the point of saturating benchmarks. A landmark achievement was the **RetinaFace** model by Deng *et al.* (2020) [3]. RetinaFace used a ResNet backbone (a deep CNN) and featured a multi-task learning head that jointly predicted face bounding boxes *and* five facial landmarks for each face [3]. This multitask design, similar in spirit to MTCNN but on a larger scale, provided the model with additional supervisory signals to refine its understanding of face geometry.

C. Vision Transformer-Based Face Detection

Vision Transformers (ViTs) represent a newer class of models that forgo convolution in favor of self-attention mechanisms to process images. Self-attention is the key breakthrough of the Transformer architecture model which enables weighing the importance of different parts of the input data (e.g., different patches in an image) against each other [12]. By calculating these relationships across the entire input, the model can capture long-range dependencies and understand global context, unlike the inherently local focus of former deep learning strategies.

Originally introduced for image classification by Dosovitskiy *et al.* in 2020 [4], the ViT model treats an image as a sequence of patch embeddings (for example, 16×16 pixel patches) and applies the Transformer architecture to learn global relationships. The success of ViTs in classification tasks—when pre-trained on sufficiently large datasets—naturally led researchers to explore their use in detection tasks.

DETR (Detection Transformer) by Carion *et al.* (2020) [2] was particularly influential in object detection, demonstrating an end-to-end Transformer-based detector that predicts objects in an image as a direct set prediction problem. DETR uses a Transformer encoder-decoder to globally reason over features and output detections directly, removing the need for mechanisms like anchor boxes (pre-defined bounding boxes to impose the scale of detected objects)[2]. DETR achieved competitive results on the COCO object detection benchmark, indicating that Transformer frameworks could handle complex detection problems. In comparison to the WIDER FACE benchmark, the COCO is a much larger-scale data set used for assessing general object detection capabilities.

Applying pure Vision Transformers (or hybrid CNN-Transformer models) to face detection is an area of ongoing research. One straightforward approach is to use a ViT as the backbone feature extractor in a standard detection pipeline. For instance, a ViT backbone can replace a ResNet in a RetinaNet or Faster R-CNN style model. The advantage is that the self-attention in ViTs can capture long-range context: a Transformer's receptive field is effectively the entire image from the start, whereas a CNN's receptive field grows with each layer. This global context could help in face detection, especially in scenes with many small faces or heavy occlusion.

Liu *et al.* (2021) developed the **Swin Transformer**, a hierarchical ViT that introduces locality by restricting self-attention to small image windows and then gradually merging patches [8]. The Swin Transformer proved very effective as a general-purpose vision backbone, achieving state-of-the-art on object detection tasks when integrated into the Mask R-CNN framework [8]. Although Swin was not designed specifically for faces, its success on detection benchmarks suggests that such ViT-based backbones can also excel at face detection when appropriately trained.

Beyond backbones, researchers have proposed Transformer-based models tailored to facial tasks. Another notable development is **MobileViT** by Mehta and Rastegari (2022), which combines CNN and ViT principles to create a lightweight hybrid model for mobile vision tasks [9]. MobileViT interleaves convolution layers with Transformer-like layers, capturing global dependencies with attention while retaining the efficient inductive biases of CNNs [9]. Hybrid CNN-ViT models like MobileViT offer a path to improved accuracy without sacrificing speed or efficiency on lightweight hardware.

It is worth noting that Vision Transformers often require large training datasets or extensive pre-training to reach their full potential in detection tasks. The WIDER FACE dataset, while large by detection standards (32,000 images with 393,703 faces) [18], is still much smaller than the datasets typically used to train ViTs from scratch [4]. Therefore, in practice, ViT-based face detectors benefit from transfer learning: starting with a model pre-trained on a broad dataset and then fine-tuning on face data.

D. Detection Accuracy

CNN-based models currently excel in face detection accuracy, particularly when trained on datasets like WIDER Face. They have undergone years of optimization for the face domain, including specialized data augmentation and multi-scale feature handling. As a result, CNN detectors like RetinaFace not only achieve high average precision (AP) on easy images but also perform strongly on challenging cases with many small, occluded faces. In contrast, Vision Transformers have shown mixed accuracy results so far in face detection. When ample training data is available, ViT-based detectors can match or even exceed CNN performance on general object detection tasks. For example, with the same ResNet backbone features, a transformer-based detector outperformed a Faster R-CNN CNN-based detector by up to 4.7 AP on COCO [12]. Similarly, using a purely transformer backbone (Swin Transformer) instead of a CNN backbone yielded higher detection precision (+3.9 AP) for similar model sizes with improved localization [12]. These results indicate the potential of transformers to offer superior accuracy by capturing global context and relationships. However, in the specific realm of face detection, CNNs still held an accuracy edge as of recent studies. It is worth noting, however, that there is limited published WIDER FACE evaluation for pure ViT models, partly because many ViT approaches to date focus on broader object detection or face *recognition* tasks.

E. Inference Speed

CNN-based models have traditionally offered strong performance in terms of inference speed. Architectures like MTCNN [19], despite their cascaded nature, can be optimized for acceptable real-time performance [15]. Single-shot detectors, including variants of models like S^{3}FD [20], are often designed with speed as a primary consideration. While highly accurate models like RetinaFace [3] provide excellent detection, comparative studies sometimes show them to be slower than highly optimized single-shot CNNs specifically tuned for speed [15]. Furthermore, lightweight CNN architectures have been developed specifically for resource-constrained environments, achieving high frame rates even on CPU hardware.

Vision Transformers, conversely, often face challenges with inference speed due to the computational complexity of their self-attention mechanisms. While DETR [2] provides an elegant end-to-end detection framework, its initial implementations generally demonstrated lower inference speeds compared to optimized CNN counterparts [14]. Similarly, while Swin Transformer [8] improves efficiency over standard ViTs, its application, particularly to detecting small faces, can still be computationally demanding compared to specialized CNNs. Hybrid models like MobileViT [9] explicitly attempt to mitigate this by combining convolutional efficiency with Transformer modeling power.

F. Conclusion

Face detection technology has advanced from hand-engineered cascades to sophisticated deep learning models in a relatively short time. CNN-based detectors have proven remarkably effective, leveraging layered convolutional features and architectural innovations to handle the challenges of scale, pose, and occlusion in face images. They currently represent the state-of-the-art in accuracy and efficiency, as evidenced by their dominant performance on benchmarks like WIDER FACE. On the other hand, Vision Transformers offer a compelling new direction, introducing the ability to capture global relationships and integrate information across an entire image via self-attention. While pure ViT models for face detection are still emerging, early results and analogies from general object detection suggest they can eventually rival CNNs, especially as large-scale pre-training becomes more common and as model architectures are refined for efficiency. In practice, hybrid models combining CNN and Transformer components are already showing the benefits of both approaches, pointing toward a future where face detectors are more accurate, robust, and adaptable than ever.

II. ML Methodology and Data Collection

A. Data Curation and Acquisition

As discussed in our literature review, vision transformers require an extremely large amount of data in order to reach their full potential. Because of this, a common approach is to train the ViT on large amounts of broad object data first, then later feed it data specific to the task at hand. We discuss this further in the future objectives section.

The data used for training our models was gathered from a variety of sources, and consists of a variety of different styles of image. In total, roughly 18,000 images were collected for potential use as training data. The majority of these images came from datasets designed for facial recognition models, such as the Labelled Faces in the Wild (LFW) Dataset [7]. So, most of the images were people facing roughly towards the camera. These datasets filled out most of our "front facing" labeled data.

Another category of data we found was images of people meant for the construction of 3D facial models, such as the Head Pose Image Database [5]. This data was great for providing us with lots of non-front-facing data, with each image being specifically labeled with the subject's orientation angle. This however, was the only data we collected that was "labeled" for our needs.

In order to begin training the models, a subset of the data was manually classified, treating all forward facing images as "detectable" and all images facing away from the camera as "not detectable". These would become the designated output for binary classification. This subset wound up being 3600 images, and served as the basis for training our model. Of the 3600, 1500 were in the "detectable" category, while 2000 were "not detectable".

B. Convolutional Neural Network Overview

Convolutional Neural Networks (CNNs) are a specialized type of neural network, optimized for processing grid-like data. This allows them to excel in fields like image processing. What sets CNNs apart from other neural network techniques, such as Multilayer Perceptrons (MLPs), is the addition of a convolutional layer.

C. Convolution Layers and kernels

A convolutional layer takes a set of weights, known in this context as a kernel, and applies them to the input data. The key here is that the kernel is many times smaller than the image data, and is applied in a sweeping-like motion across the input data. This process greatly reduces the dimensionality of the given data and is a key aspect of CNNs. This reduced dimensionality has many benefits. One of these is an increase in efficiency, as there is less data to process after the convolutional layer. It also benefits the model by removing unnecessary information from the data, highlighting only the important information. Through backpropagation, the kernel is adjusted to highlight the parts of the image that produce the best results, increasing the accuracy of the model. When choosing the starting kernel for a particular neural network, there are known patterns that affect images in predictable ways. When combined with some domain knowledge and insight, picking the right starting values for a kernel can lower the training time needed to hit certain levels of accuracy. These kernels are optimized for things like edge detection, image sharpening, and image smoothing.

D. Pooling and Dropout

The other techniques used within our CNN include pooling and dropout. Pooling refers to the process of reducing the dimension of a section of data while retaining the most important information. We made use of the most common form, max pooling. In max pooling only the greatest value within the pooled section is kept, and the other values are discarded. This is particularly useful for images with very large starting dimensions. Dropout is a technique used primarily to reduce overfitting in neural networks.[17] For each training step, a given percentage of weights within a model are zeroed out. What this does is force the neurons of the model to be less reliant on their neighboring weights, and form more independent and complex connections. This forcing each neuron to "carry more weight" leads to an individual performance increase, which in turn leads to an increase in the overall layer's accuracy. Dropout is not done during the model validation, allowing the model to use all of its learned parameters. It is still unclear what the effects of dropout are on convolutional layers, however, so it was not added to ours. [16]

E. Adam Optimizer

In trying different optimizers, we found the most success when training with the Adam optimizer function. This method is based on stochastic gradient descent and, among other things, differs by factors in a model's momentum. In the context of neural networks, momentum is a technique to improve upon regular gradient descent. It works by increasing the speed at which gradient descent takes place by incorporating past gradient updates into the gradient calculation, allowing for efficient and more accurate weight updates. [13]

F. Vision Transformer Overview

Vision transformers (ViTs) represent the beginning of a shift in computer vision techniques by using the transformer architecture. This style of architecture was originally designed for natural language processing, and its success has caused people to use it in more diverse ways. Unlike conventional convolution-based systems that process an entire image holistically, our implementation breaks the image into discrete patches and processes them sequentially like processing words in a sentence.

1) Patch Embedding and Tokenization: Our model first divides the input image into non-overlapping patches using a convolution operation. Then, it projects each patch into a higher-dimensional vector space with linear projection, converting the visual information into a numerical representation the computer can interpret.

In addition to the numerical representation of the patches, a classification (CLS) token is pre-pended to the sequence. The CLS token does not represent any single patch, but rather serves as a summary of the results as the model progresses through the layers, allowing the model to compile global relationships. The model ultimately uses the CLS token to make its final class prediction.

2) Spatial Information: Since the transformer architecture inherently does not recognize original ordering or sequence location, our implementation augments each token with a learnable positional embedding. These embeddings encode the spatial layout of the patches, ensuring that the positional context (i.e., where a patch is located in the original image) is maintained.

3) Transformer Blocks: The sequence of enriched tokens is then forwarded through a series of transformer encoder blocks. Each block is designed to progressively refine the representations using two primary submodules:

- *Multi-Head Self-Attention:* Within each transformer block, a multi-head attention mechanism allows every token to "look at" all the other tokens. This is achieved by projecting the token embeddings into three distinct sets of vectors (queries, keys, and values)[1] [2] [3]. and computing scaled dot-product attention between them. The weights for the query, key, and value projections are learnable parameters. This process enables the model to dynamically re-weight the importance of each patch in relation to every other patch, thereby capturing long-range dependencies and subtle features that contribute to facial detection.
- *Feed-Forward Network:* Following the attention step, a multi-layer perceptron applies transformations (using the GELU activation function) and additional dropout, refining the features before they are passed to the next layer.

4) Model Stability: To enhance stability and performance, we leveraged residual connections and layer normalization before each of the attention and MLP stages. We also implemented stochastic depth whereby selected residual connections are randomly dropped during training. By increasing the drop probability in deeper layers, stochastic depth acts as a regularization strategy that improves generalization over baseline ViT models and minimizes overfitting.

5) Classification: Once the sequence has passed through all transformer blocks, a final layer normalization is applied. The CLS token, which has been aggregating global information throughout the network, is then extracted from the sequence. This token is fed into a linear classification head that predicts whether a face is or is not detectable.

6) Our Model's Differences: Compared to standard vision transformer models, our implementation offers several key enhancements:

- *Adaptive Regularization:* The use of stochastic depth minimizes overfitting and promotes the learning of robust representations, especially in deeper networks.
- *Learnable Positional and Class Tokens:* The incorporation of learnable tokens ensures that both global and spatial contexts are preserved, improving the model's capacity to detect facial structures even under variations in scale and orientation.
- *Convolution-Based Patch Extraction* By using a convolution layer for patch embedding, our approach efficiently captures local image statistics, which is particularly beneficial for facial features that are spatially localized.

III. Model Results & Interpretation

A. CNN Results

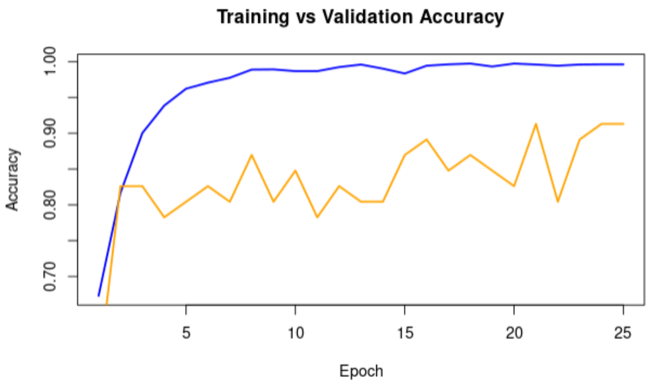

Fig. 1. The best performing CNN model achieved an accuracy of 91.3% on the validation data after 25 epochs. The near 100% training accuracy and jumpy validation accuracy implies some overfitting, but overall the model accurately

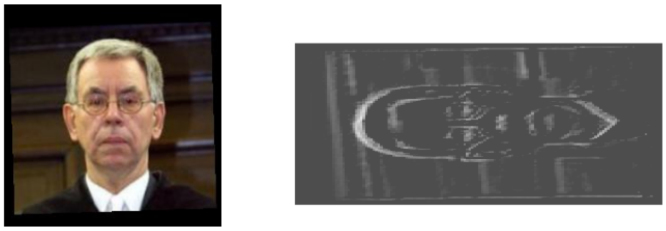

Fig. 2. A visualization of an image before and after our first convolutional layer

B. CNN Analysis

After experimenting with a variety of different network architectures, hyper parameters, loss functions, optimizers, and more, we achieved a CNN with a validation accuracy of 91.3%. These results are consistent with CNN's current status as the dominant model in the image processing space. Even with a somewhat small dataset and a complex task, the CNN model was able to achieve a high success rate, with what appears to be only a small amount of overfitting.

1) Final Architecture: This model consisted of two convolutional layers, a dense layer, and an output layer. The first convolution layer made use of a 5x5 kernel, 32 filters, a ReLU activation function, and was followed by a 2x2 max pooling layer. The second convolution layer differed only in that it used a 3x3 kernel and 64 filters. The convolution layers were followed by a dense layer with 256 nodes, also making use of the ReLU activation function. We found that our dense hidden layer, when coupled with a dropout step with a dropout rate of 0.5 to be the most performate strategy. The output layer was set up for binary classification with a sigmoid activation function. For our lost function, binary cross entropy was used. As discussed in our methodology, the Adam optimizer gave us the best outcomes when compared to other methods like Stochastic Gradient Descent.

2) Kernel Analysis: As seen in Figure 2, our first convolutional layer seems to take the approach of highlighting important facial characteristics as well as the edges of the face in the photo. This provides insight into how our model actually "recognizes" faces, and shows how it detects some of the features you would expect a facial detection model to call out.

[1] **Query:** A query is a vector representation computed from an input element that specifies what related information it seeks from other elements.

[2] **Key:** A key is a vector representation computed from an input element that serves as an identifier, allowing the model to assess how relevant this element is to a given query.

[3] **Value:** A value is a vector representation computed from an input element that contains the actual information or features, which is then aggregated based on its relevance to the query.

C. *ViT Results*

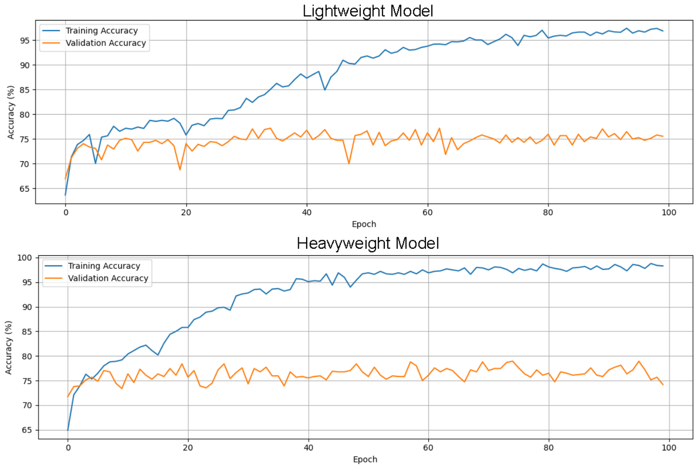

Fig. 3. Comparison between lightweight ViT (top) and heavyweight ViT (bottom)

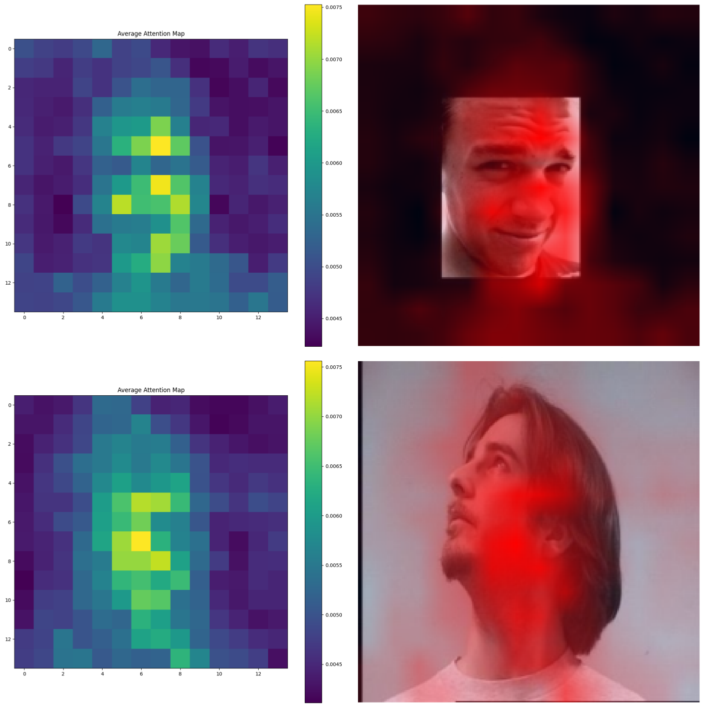

Fig. 4. Heavyweight model's attention visualized on positive (top) and negative (bottom) labels

D. *ViT Analysis*

1) Accuracy & Overfitting: We evaluated two Vision Transformer (ViT) models with varying complexities for the face detection task: a lightweight model and a heavyweight model. The lightweight model was configured with an embedding dimension of 512, 8 attention heads, and 6 transformer blocks. The heavyweight model utilized an embedding dimension of 768, 12 attention heads, and 12 transformer blocks.

Both models exhibited rapid learning, achieving approximately 72% accuracy on the validation set within the first 5 epochs. However, both models also demonstrated severe overfitting. Specifically, the lightweight model reached a peak validation accuracy of 77.17% around epoch 50, after which the validation accuracy began to decline while the training accuracy continued to increase. The heavyweight model, due to its increased capacity, overfit even more quickly, reaching a peak validation accuracy of 78.80% around epoch 30 before the divergence between training and validation performance became apparent.

These results highlight the potential and limitations of Vision Transformers for face detection. While the models achieved decently accurate detection abilities, they also underscore the importance of regularization techniques to mitigate overfitting, especially when training data is limited.

2) Attention Heat Maps: The attention heat maps revealed distinct patterns for positive (face detectable) and negative (face undetectable) images, as shown in Figure 4.

For positive-labeled images, the attention heads tended to concentrate on the facial region. The average attention map shows a clear concentration of attention in the central area of the image, corresponding to the location of the face. When overlaid on the original image, the heatmap highlights the facial features, indicating that the model is learning to attend to relevant regions for face detection.

In contrast, for negative-labeled images, the attention patterns were more scattered. The average attention map shows a weaker and more spread-out attention pattern, with no clear concentration on any specific region. When overlaid on the original image, the heatmap does not highlight any particular features, suggesting that the model is not finding any face-like patterns in these images.

It is important to note that the attention maps are not perfectly focused on specific facial features such as eyes or nose. This may be due to the variability in facial pose, expression, and lighting conditions in the dataset. Further research is needed to explore techniques for improving the precision and interpretability of attention maps in ViT models for face detection.

3) Weight Initializations: We investigated the impact of different weight initialization strategies on the performance of the ViT model. As a baseline, we initialized the weights of all layers, the classification (CLS) token, and the positional embedding vectors using a normal distribution with a standard deviation of 0.02.

We hypothesized that certain layers might benefit from more specialized initialization techniques. Specifically, we explored He initialization for the convolutional layer in the patch embedding module, inspired by the work of Nisonoff [10]. However, this approach resulted in a decrease of approximately 8% in validation accuracy during the initial epochs compared to the baseline random initialization. This suggests that He initialization, which is designed to prevent vanishing gradients in deep CNNs, may not be well-suited for the specific characteristics of the patch embedding layer in the ViT model.

We also investigated the use of sinusoidal positional embeddings, as proposed in the original Transformer paper [11]. While the model with sinusoidal embeddings eventually converged to a similar validation accuracy as the baseline model after approximately 9 epochs, it exhibited a 4% lower validation accuracy during the initial epochs. This indicates that learning positional embeddings may be more effective for capturing the specific positional relationships in the face detection task, at least during the early stages of training.

IV. Future Work

A. *CNN Improvements*

Our CNN model was able to successfully differentiate between images in terms of facial orientation detection. Regardless, there are still areas for improvement that can increase the overall reliability and accuracy of the model, and potentially decrease some of the slight overfitting that appears to be happening. Two potential areas for improvement are defining a starter kernel and adding synthetic data through data augmentation.

1) Starting training with a preset kernel: For convolutional layers, the kernel is what dictates what is passed down to the rest of the model. The layers within a neural network are generally "black boxes", while we don't know the exact values for what is happening within them, the initial weights are free to be initialized however. For our implementation, we chose to start with random weights, but a strategy that could improve out results is to begin training our model with a preset kernel. There are known kernel arrangements that produce different results on the given data, some resulting in overall sharpening, edge sharpening, image smoothing, and more. Based on the kernels our model naturally produced, starting training with a kernel designed to sharpen the edges of the content within an image may increase the accuracy of our results and the speed at which we get them.

2) Increasing our dataset through data augmentation: Another potential for model improvement comes from an increased quantity of training data. More data of good quality is always useful for models, especially for those that both suffer from overfitting. However, because the majority of publicly available data isn't classified for facial orientation, classifying data must be done manually or with other models. In order to avoid some of the pitfalls that come from this, a common strategy is to augment your existing classified data in order to artificially produce more novel data. If done well, this augmented data is different enough from the date it was produced from, and can be used to train the model without the risk of additional overfitting.

C. ViT Improvements

While our Vision Transformer (ViT) models demonstrated rapid initial learning, their performance was ultimately impacted by severe overfitting, likely due to the data-hungry nature of transformers. Future work should prioritize strategies to mitigate this overfitting and leverage the unique strengths of the transformer architecture more effectively for face detection. We propose focusing on the following areas:

3) Enhanced Pre-training and Transfer Learning Strategies: The standard practice for ViTs involves pre-training on massive datasets before fine-tuning on a specific task. Future work should explore:

- **Diverse Pre-training Datasets:** Instead of relying solely on image classification pre-training, investigating pre-training on large-scale object detection datasets (like COCO) could be beneficial.
- **Self-Supervised Pre-training:** Exploring self-supervised learning (SSL) methods for pre-training could allow the model to learn rich visual representations from unlabeled data. This may reduce the necessity of massive labeled datasets and lead to more generalizable features.

4) Advanced Regularization and Data Augmentation: Given the pronounced overfitting observed, especially in the heavyweight ViT model, implementing more sophisticated regularization techniques beyond the stochastic depth used is essential. Future iterations should investigate:

- **Stronger Augmentation Policies:** Implementing advanced data augmentation techniques specifically beneficial for ViTs, such as *CutMix* (cutting and pasting patches between images) [1]. These methods act as powerful regularizers by preventing the model from simply memorizing training examples and encouraging it to learn more robust, invariant features.
- **Targeted Regularization Techniques:** Experimenting with different dropout rates within the attention and feed-forward layers or increasing weight decay could further combat overfitting.
- **Model Architecture Adjustment for Regularization:** Explicitly exploring hyperparameters related to model complexity, such as reducing the embedding dimension, the number of attention heads, or the network depth, could directly trade off some model capacity for improved generalization, finding a better balance for the available data size.

5) Exploring Hybrid CNN-ViT Architectures: The literature review highlighted promising hybrid models (like MobileViT) that combine the strengths of CNNs (efficiency, strong local feature extraction) and ViTs (global context modeling). Instead of treating CNNs and ViTs as entirely separate approaches, future work could focus on hybrid integration:

- **Convolutional Stem:** While our current ViT uses a convolutional layer for patch embedding, exploring more sophisticated convolutional "stems" (multiple initial CNN layers) could provide richer local features as input to the transformer blocks.
- **Integrating Convolution within Transformer Blocks:** Investigating architectures that interleave convolutional layers with self-attention layers within the main transformer body. This could help the model maintain strong local inductive biases while still benefiting from the global context provided by attention, potentially offering a better trade-off between performance, efficiency, and data requirements compared to pure ViT models.

References

[1] Sihun Baek et al. *Visual Transformer Meets CutMix for Improved Accuracy, Communication Efficiency, and Data Privacy in Split Learning*. 2022. arXiv: 2207.00234 [cs.LG]. URL: https://arxiv.org/abs/2207.00234.

[2] Nicolas Carion et al. "End-to-end object detection with transformers". In: *Proc. ECCV*. 2020, pp. 213–229. URL: https://arxiv.org/abs/2005.12872.

[3] Jiankang Deng et al. "RetinaFace: Single-Shot Multi-Level Face Localisation in the Wild". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 5202–5211. DOI: 10.1109/CVPR42600.2020.00525.

[4] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *Proc. ICLR*. 2021. URL: https://arxiv.org/abs/2010.11929.

[5] Nicolas Gourier, Daniela Hall, and James Crowley. *Estimating Face orientation from Robust Detection of Salient Facial Structures*. URL: http://crowley-coutaz.fr/jlc/papers/Pointing04-Gourier.pdf (visited on 04/15/2025).

[6] Haoxiang Li et al. "A convolutional neural network cascade for face detection". In: *Proc. CVPR*. 2015, pp. 5325–5334. URL: https://openaccess.thecvf.com/content_cvpr_2015/papers/Li_A_Convolutional_Neural_2015_CVPR_paper.pdf.

[7] Jessica Li. *Labelled Faces in the Wild (LFW) Dataset*. www.kaggle.com, 2018. URL: https://www.kaggle.com/datasets/jessicali9530/lfw-dataset.

[8] Ze Liu et al. “Swin Transformer: Hierarchical vision transformer using shifted windows”. In: *Proc. ICCV*. 2021, pp. 10012–10022. URL: https://arxiv.org/abs/2103.14030.

[9] Sachin Mehta and Mohammad Rastegari. “MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer”. In: *Proc. ICLR*. 2022. URL: https://arxiv.org/abs/2110.02178.

[10] Tyler Nisonoff. *Weight initialization for cnns: A deep dive into he initialization*. 2018. URL: https://medium.com/@tylernisonoff/weight-initialization-for-cnns-a-deep-dive-into-he-initialization-50b03f37f53d.

[11] Nikhil Chowdary Paleti. *Positional encoding explained: A deep dive into transformer PE*. 2024. URL: https://medium.com/thedeephub/positional-encoding-explained-a-deep-dive-into-transformer-pe-65cfe8cfe10b.

[12] Picsellia. *Are Transformers Replacing CNNs in Object Detection?* https://www.picsellia.com/post/are-transformers-replacing-cnns-in-object-detection. Accessed: March 29, 2025.

[13] Arnab Sinha. *Momentum: A simple, yet efficient optimizing technique*. June 2019. URL: https://medium.com/analytics-vidhya/momentum-a-simple-yet-efficient-optimizing-technique-ef76834e4423 (visited on 04/15/2025).

[14] Huaiyuan Sun et al. “Pruning DETR: efficient end-to-end object detection with sparse structured pruning”. In: *Signal, Image and Video Processing* 18 (2023), pp. 129–135. URL: https://api.semanticscholar.org/CorpusID:264936349.

[15] Sumit Tariyal et al. “A comparitive study of MTCNN, Viola-Jones, SSD and YOLO face detection algorithms”. In: *2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE)*. 2024, pp. 1–7. DOI: 10.1109/IITCEE59897.2024.10467445.

[16] Haibing Wu and Xiaodong Gu. “Towards dropout training for convolutional neural networks”. In: *Neural Networks* 71 (Nov. 2015), pp. 1–10. DOI: 10.1016/j.neunet.2015.07.007. URL: https://cs.nju.edu.cn/wujx/paper/CNN.pdf.

[17] Harsh Yadav. *Dropout in Neural Networks | Towards Data Science*. Towards Data Science, July 2022. URL: https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9/.

[18] Shuo Yang et al. “WIDER FACE: A Face Detection Benchmark”. In: *CoRR* abs/1511.06523 (2015). arXiv: 1511.06523. URL: http://arxiv.org/abs/1511.06523.

[19] Kaipeng Zhang et al. “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”. In: *CoRR* abs/1604.02878 (2016). arXiv: 1604.02878. URL: http://arxiv.org/abs/1604.02878.

[20] Shifeng Zhang et al. “S^{3}FD: Single Shot Scale-invariant Face Detector”. In: *CoRR* abs/1708.05237 (2017). arXiv: 1708.05237. URL: http://arxiv.org/abs/1708.05237.