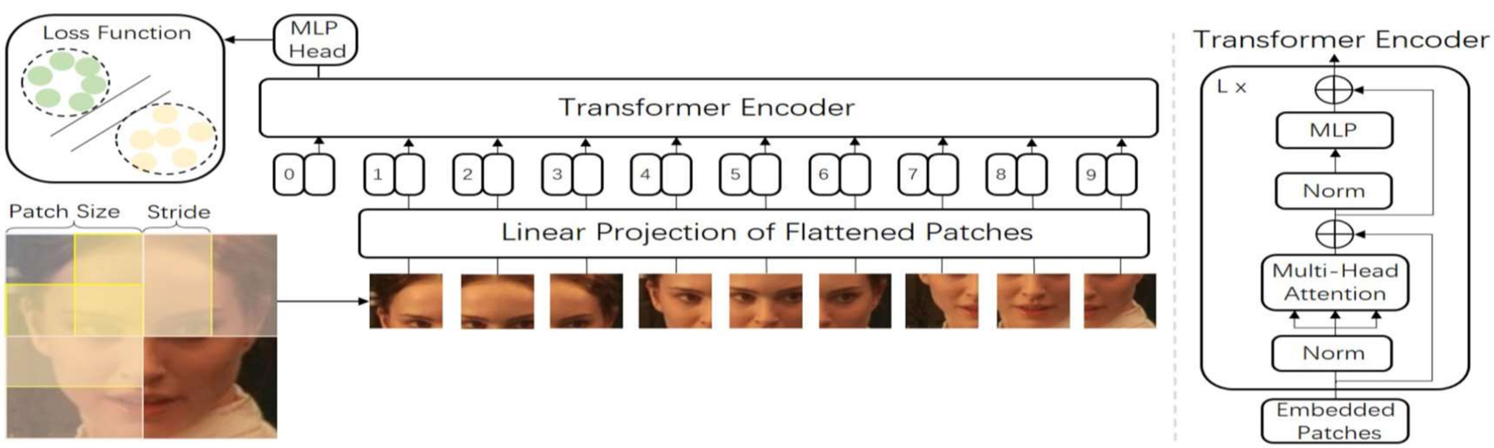
Face Transformer

Rethinking model incorporating EfficientNet into ViT

INTRODUCTION

Recently there has been a growing interest in Transformer not only in **NLP** but also in **Computer Vision**. The transformer can be used in **Face Recognition** and it is better than CNNs. Therefore, we investigate the performance of Transformer models in Face Recognition. Considering the original Transformer may neglect the interpatch information, we modify the patch generation process and make the tokens with sliding patches that overlap with each other. The models are trained on **CASIA-WebFace** databases, and evaluated on several mainstream benchmarks, including **LFW** databases. We demonstrate that Face Transformer models trained on a large-scale database, **CASIA-WebFace**, achieve comparable performance as CNN with a similar number of parameters and MACs. The Face-Transformer mainly uses **ViT (Vision Transformer)** architecture. Now we demonstrate if we can transfer learn and fine-tune the model with **EfficientNet** & merge it into **ViT** to get a better result.

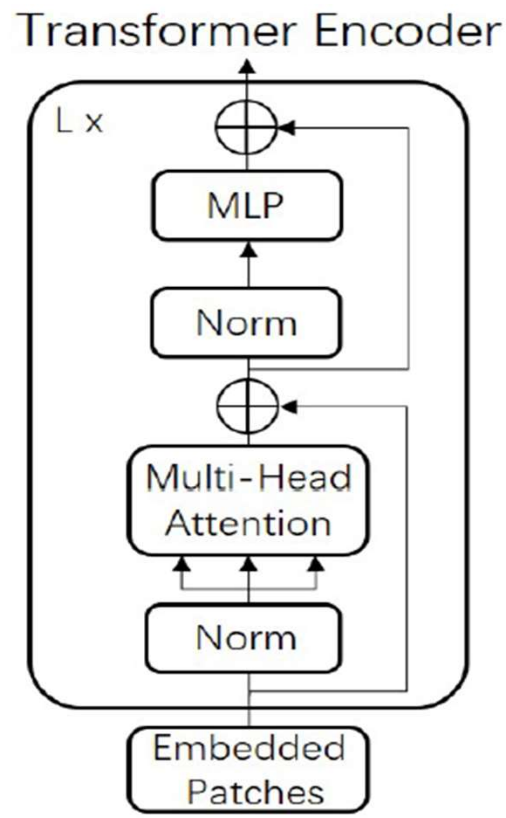


What is Transformer?

- A transformer is a type of deep learning model that uses Artificial Neural Networks to process sequential input data. Transformers are used for Natural Language Processing (NLP) and Computer Vision (CV) tasks.

- Transformers use a mechanism called **Self-Attention** to process input data. This mechanism helps identify how distant data elements influence and depend on one another. Transformers can process the entire input data at once, capturing context and relevance.

- Transformers can capture long-range dependencies and relationships between patches in the image more effectively by using self-attention rather than convolutions.



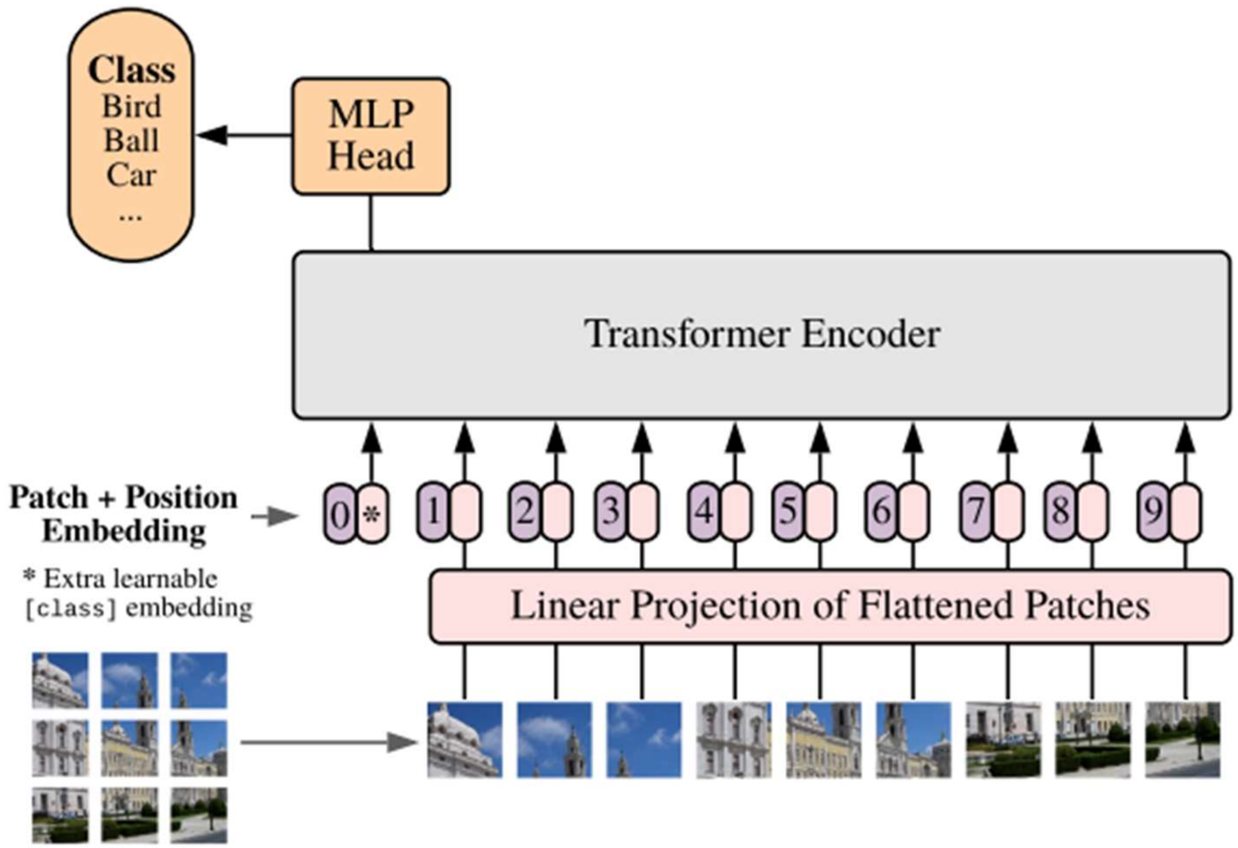
Vision Transformer (ViT)

- A Vision Transformer (ViT) is a type of neural network architecture for computer vision tasks that utilizes the transformer architecture, originally introduced for natural language processing and computer vision tasks.

- Vision Transformer (ViT) is a type of neural network architecture for computer vision tasks like image classification, object detection, and image segmentation. It takes inspiration from the Face Evolve Model - https://github.com/ZhaoJ9014/face.evolve, a high-performance **Face Recognition Library** based on **PaddlePaddle** & **PyTorch.**

- **Face Transformer** for Recognition is a specific deep learning architecture designed for facial recognition tasks, inspired by ViT and Face-Evolve. It represents a novel approach that merges the strengths of traditional Convolutional Neural Networks (CNNs) with the powerful self-attention capabilities of Transformers.

Vision Transformer (ViT)



Salient Features of Vision Transformer

Features	Description
Self-Attention Mechanism	ViT eschews the traditional convolutional layers of CNNs and instead relies on the powerful self-attention mechanism. This mechanism allows each patch in an image to "attend" to all other patches, effectively analyzing their relationships and dependencies
Patch-based Processing	Instead of operating on the entire image at once, ViT divides it into smaller, overlapping patches. Analyzing smaller patches is computationally cheaper than processing the entire image, making ViT potentially more efficient.
Intoducing Loss Functions	ViT introduces cross entropy based loss functions like CosFace, ArcFace, SFaceLoss, SoftMax etc.
Local and global feature extraction	Patches capture local details, while their overlap allows interaction and understanding of global relationships.

Objectives

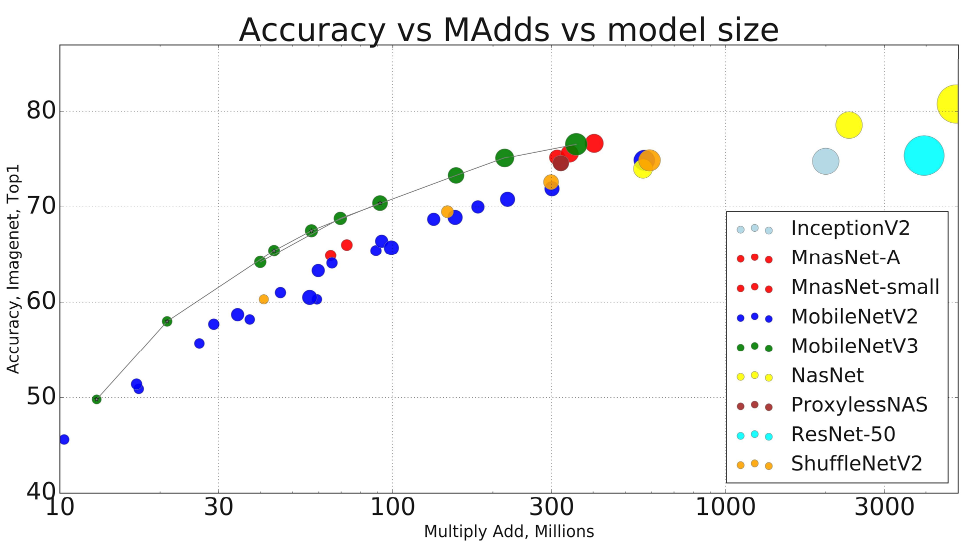
- To learn a representation of face images that is invariant to variations in lighting, pose, and expression.
- To achieve state-of-the-art results on face recognition benchmarks by fine-tuning with EfficientNet and introduce the model into ViT.
- To be robust to variations in the quality of the input images by evaluating **LFW** evaluation databases.
- To make it efficient in terms of computational cost and memory.

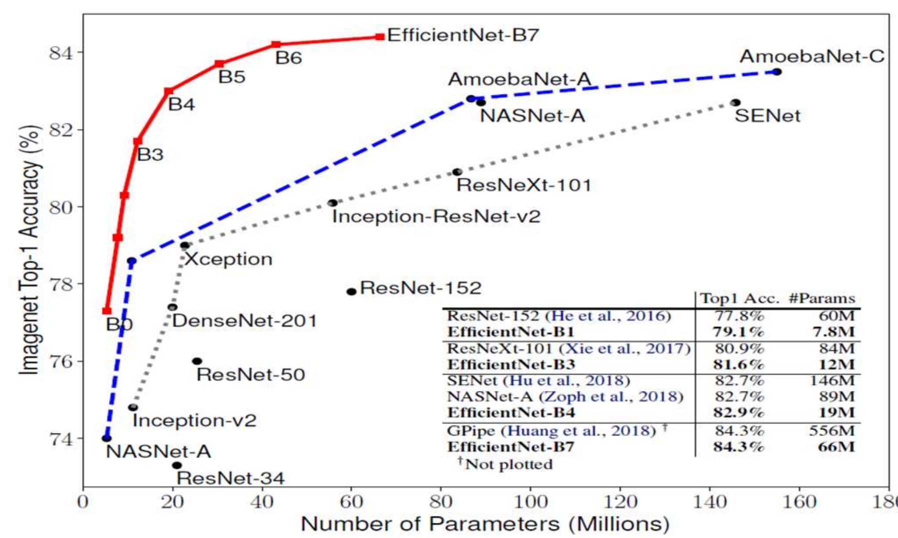
Proposed Solution

Solutions	Description
Using more powerful hardware	Face transformer models can be made more computationally efficient by using more powerful hardware, such as GPUs and TPUs
Using more advanced techniques	Transfer Learning and Fine-Tuning through EfficientNet & ViT.
Collecting more data	The performance of face transformer models could also be improved by collecting more data. This could be done by collecting data from a wider variety of sources.

Why EfficientNet?

Two successful model of computer vision (CV) related works are – **MobileNetV1 (2017), MobileNetV2 (2018), MobileNetV3 (2019)** & **EfficientNetV1 (2019), EfficientNetV2 (2021)**. Convolutional Neural Networks (ConvNets) are commonly developed at a fixed resource budget and then scaled up for better accuracy if more resources are available. EfficientNet systematically studies model scaling and identifies that carefully balancing network depth (d), width (w), and resolution (r) can lead to better performance. Neural architecture searches to design a new baseline network and scale it up to obtain a family of models, called EfficientNets, which achieve much better accuracy and efficiency than previous ConvNets. In particular, our EfficientNet achieves state-of-the-art **84.3% top-1 accuracy** on ImageNet, while being **8.4x smaller** and **6.1x faster** on inference than the best existing ConvNet.

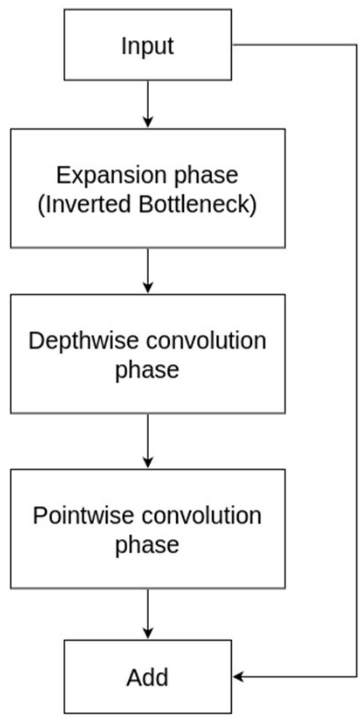




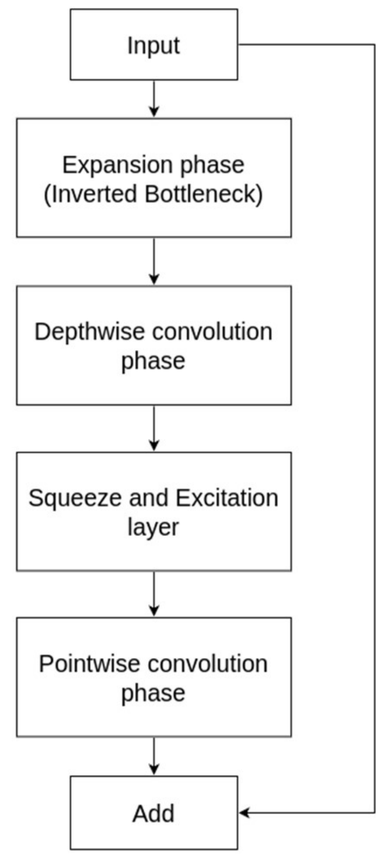
MobileNetV2 vs EfficientNet

Aspect	MobileNetV2	EfficientNet
Architecture	Depthwise separable convolutions, linear bottlenecks	Uniform scaling of depth, width, and resolution
Efficiency	Efficient for mobile and embedded vision applications	Achieves high accuracy while maintaining efficiency
Performance	Balanced accuracy, speed, and model size	Generally outperforms MobileNetV2 on benchmarks
Model Size	Smaller and more lightweight	Larger, but offers state-of-the-art performance
Computational Resources	Less computationally intensive	More computationally intensive, but efficient scaling
Deployment	Suitable for resource-constrained environments	Better suited when slightly larger models can be accommodated
Established	Longer established, widely used	Relatively newer but rapidly gaining popularity

MobileNetV2 vs EfficientNet

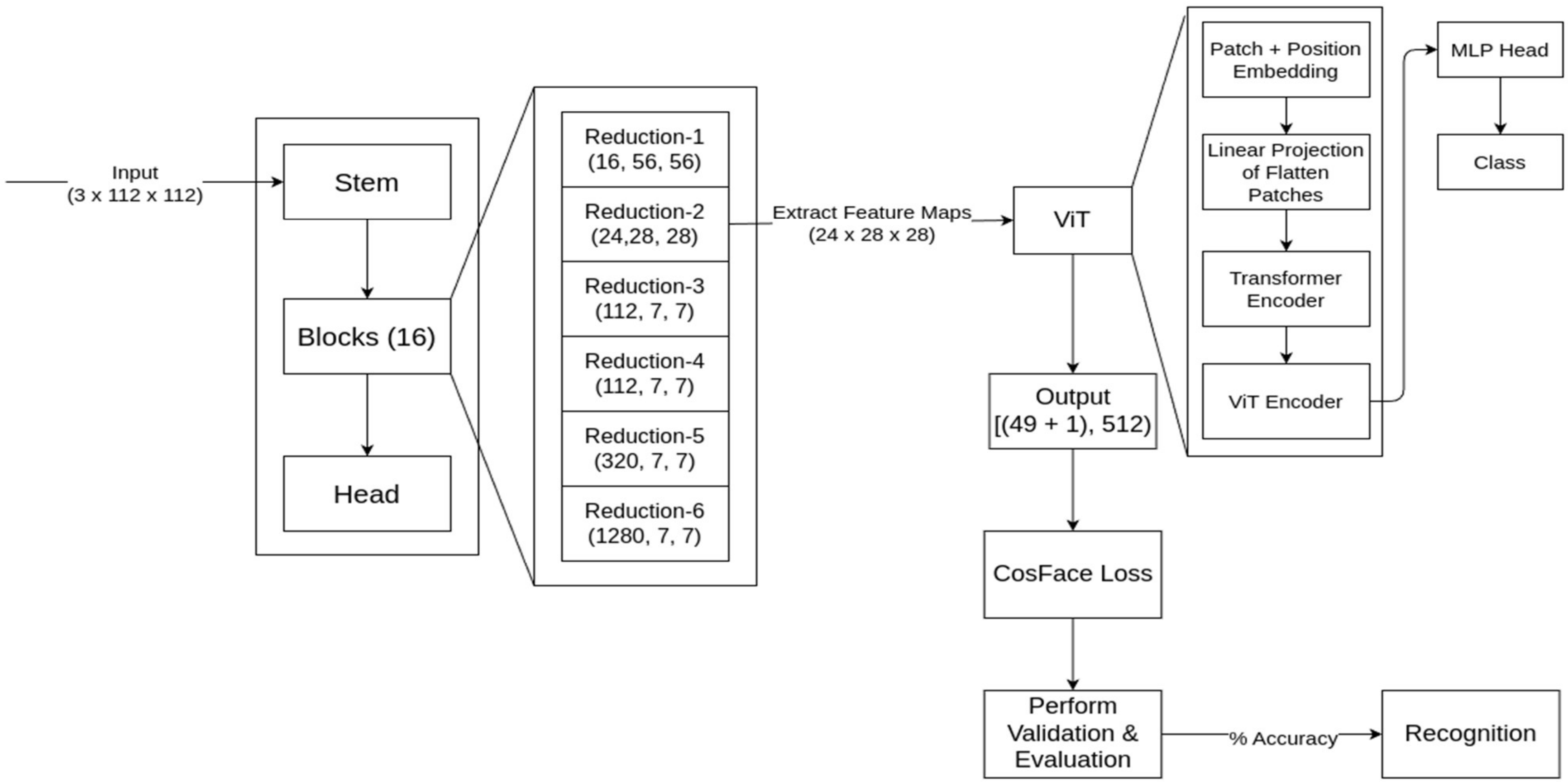


MobileNetV2



EfficientNet

Our Model Architecture



Explanation of Model Architecture

- In the block structure there exists 6 reduction block 1 to 6 respectively extracted from the block level. The reduction layer architecture shown on the below Table 3.3

- From the reduction-2 layer that gives the spatial dimension of $24 \times 28 \times 28$ for an input size of $3 \times 112 \times 112$, we extract the feature maps.

- Then the extracted feature maps are passed through ViT and gives the output size of [(49 + 1), 512], here 1 refers to the `[cls]` tokenizer & 512 refers to the Embedding size.

- Then we introduce the predefined loss function CosFace Loss & perform evaluation on the output model, that gives us the percentage accuracy.

- After running for nearly 7200 batches for 16 epochs each and , our model achieves nearly 96.41% accuracy on LFW Evaluation. We expect that, for a long time of training and evaluation as well as a single run can achieve a greater accuracy for the model.

- **NOTE:** The learning rate have decided 3×10^{-5} , as neither 10^{-4} nor 10^{-6} can not achieve the convergence. For 10^{-4}, there has been a problem of non-convergence and for 10^{-6} , the learning rate stuck at a fixed accuracy of 50% & it does not show any improvement.

Stage	Operator	Resolution	#Channels	#Layers
i	F_i	$H_i \times W_i$	C_i	L_i
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	28×28	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Table 3.1: EfficientNet-B0 baseline network - Each row describes a stage i with L_i layers, with input resolution (H_i, W_i) and output channels C_i. For spatial dimension of 224×224

Stage	Operator	Resolution	#Channels	#Layers
i	F_i	$H_i \times W_i$	C_i	L_i
1	Conv3x3	112×112	32	1
2	MBConv1, k3x3	56×56	16	1
3	MBConv6, k3x3	56×56	24	2
4	MBConv6, k5x5	28×28	40	2
5	MBConv6, k3x3	14×14	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	7×7	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

Table 3.2: EfficientNet-B0 baseline network - Each row describes a stage i with L_i layers, with input resolution (H_i, W_i) and output channels C_i. For spatial dimension of 112×112.

Reduction Level	**Shape**	
	when spatial dimension = 224×224	**when spatial dimension = 112×112**
endpoint[reduction 1]	(1,16,112,112)	(1,16,56,56)
endpoint[reduction 2]	(1,24,56,56)	(1,24,28,28)
endpoint[reduction 3]	(1,40,28,28)	(1,40,14,14)
endpoint[reduction 4]	(1,112,14,14)	(1,112,7,7)
endpoint[reduction 5]	(1,320,7,7)	(1,320,7,7)
endpoint[reduction 6]	(1,1280,7,7)	(1,1280,7,7)

Table 3.3: Reduction Table

Results & Output

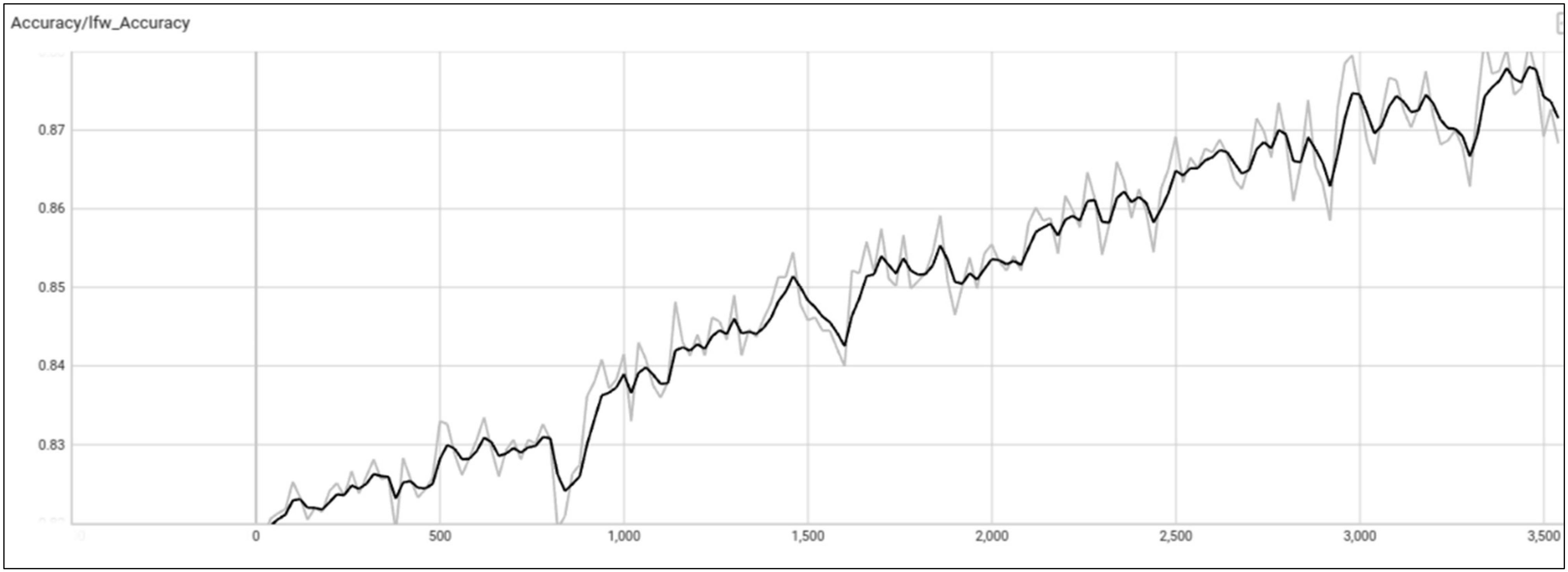
Training Data	Model	LFW	SLLFW	CALFW	CPLFW	TALFW	CFP-FP	AGEDB-30
CASIA-WebFace	ViT-P8S8	97.32%	90.78%	86.78%	80.78%	83.05%	86.60%	81.48%
	ViT-P12S8	97.42%	90.07%	87.35%	81.60%	84.00%	85.56%	81.48%
	EfficientNet	92.73%	-	-	-	-	-	-
	EfficientNet + ViT	96.41%	-	-	-	-	-	-

Table 4.1: Performance on LFW, SLLFW, CALFW, CPLFW, TALFW, CFP-FP & AGEDB-30 Databases

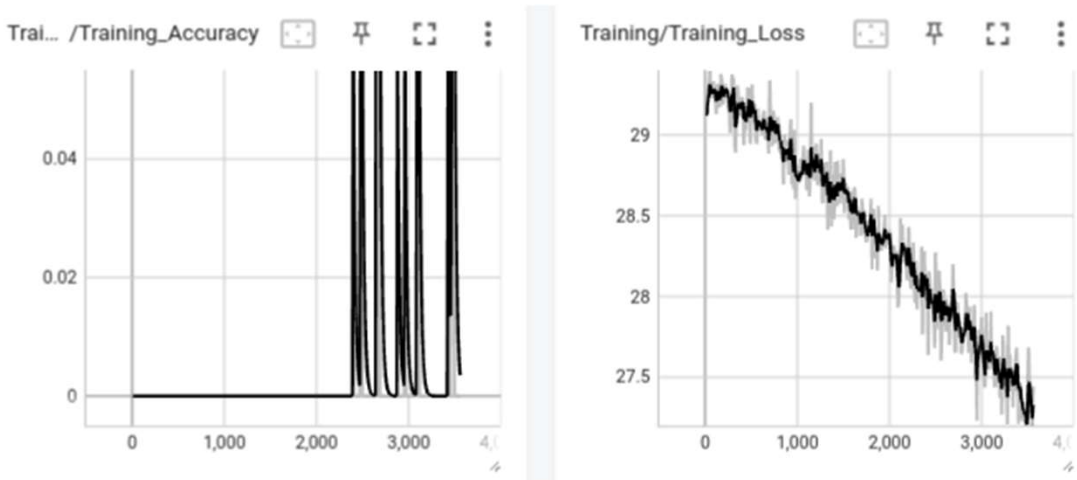
Results & Output

Model Name	Training Data	Accuracy	No. of days taken to reach the accuracy
ViT-P8S8	CASIA-WebFace	97.32%	37 Days
ViT-P12S8		97.42%	36 Days
EfficientNet		92.73%	9 Days
EfficientNet + ViT		96.41%	14 Days

Table 4.2: Time taken to train with the described model

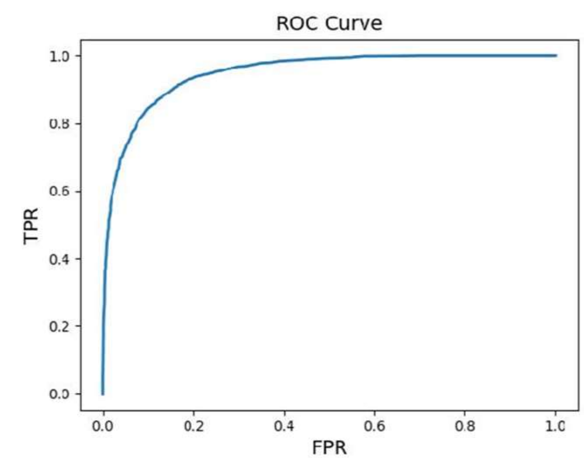


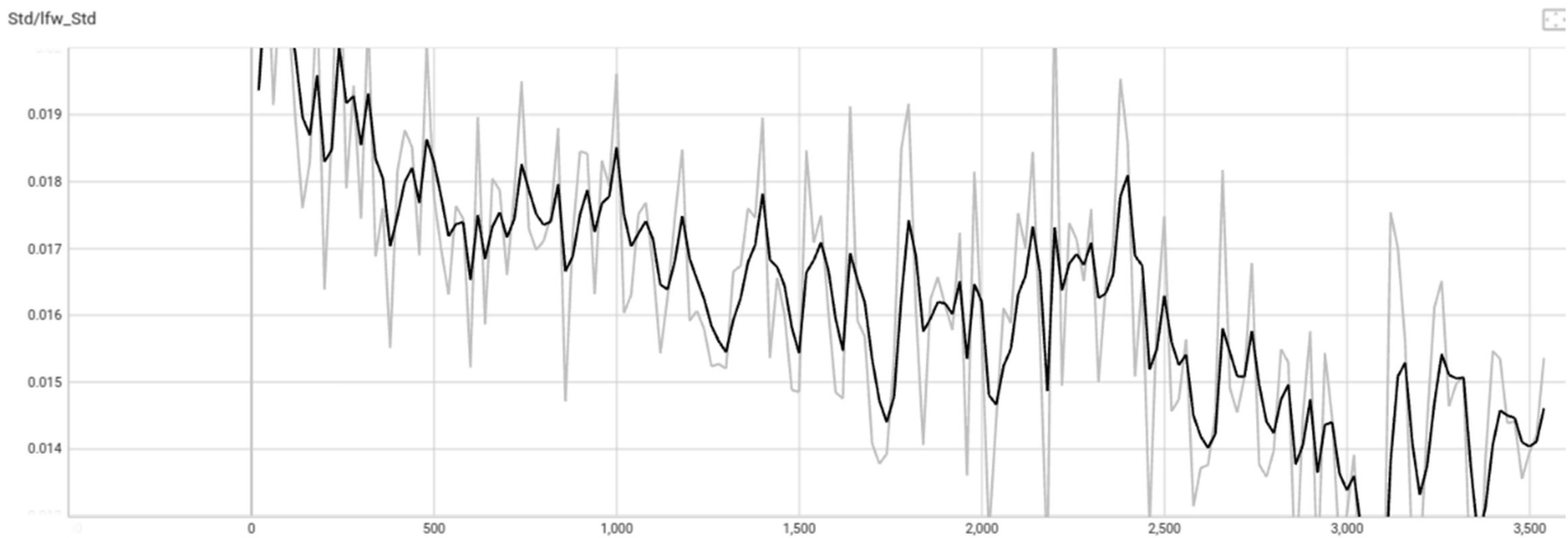
Accuracy / lfw-Accuracy



Training Accuracy & Training Loss

ROC Curve




Std/lfw_Std
0.019
0.018
0.017
0.016
0.015
0.014
0
500
1,000
1,500
2,000
2,500
3,000
3,500

Std / lfw_Std

XNorm / lfw-XNorm

Conclusion

Despite encountering hardware issues we run the project in Google Colab using saved checkpoint, due to the computational limitations in Google Colab. After running for nearly 7200 batches for 16 epochs each, our model achieves nearly 96.41% accuracy on LFW Evaluation, where the ViT accuracy reaches upto 97.32% for a prolong computational run of 37 days. That means our experiment of transfer learn a model with EfficientNet & merged it into ViT, making the model EfficientViT becomes successful enough. Our goal is to unlock the full potential of the collaborative model and deliver performance closer to, or even surpassing, the merged capabilities of EfficientNet and Vision Transformer (ViT).

With the completion of our project, the success of our EfficientViT model stands as a testament to our dedication. Having merged the strengths of EfficientNet and ViT, we've achieved remarkable results, paving the way for future advancements in computer vision. As we reflect on our journey, we're proud to have surpassed the capabilities of our individual models and delivered performance beyond expectations.