National University of Computer & Emerging Sciences

FACE RECOGNITION

Swin & Vision Transformers on LFW Dataset

PROJECT REPORT

Project Team:

- K21-4579 - Muhammad Hamza
- K21-4871 - Emmanuel
- K21-3204 - Jatin Kesnani

Project Repository: face-recognition-vit-and-swin

Lecturer: Ms. Sumaiyah Zahid

Course: Deep Learning for Perception (CS - 4045)

Section: BCS - 8A & B

Semester: Spring 2025

Department: Department of Computer Science

Campus: Karachi, Sindh, Pakistan

Submission Date: May 4, 2025

Face Recognition Using Swin and Vision Transformers on LFW

Abstract

This report presents a comprehensive study on face recognition employing two state-of-the-art transformer architectures: the Vision Transformer (ViT) [2] and the Swin Transformer [3]. Both models are trained and evaluated on the Labeled Faces in the Wild (LFW) dataset [1]. We detail data preprocessing, model architectures, training protocols, and comparative results in terms of accuracy, ROC AUC, and computational efficiency. Extensive code listings and clear citations throughout provide reproducibility and rigor.

***Index Terms*—Face Recognition, Vision Transformer, Swin Transformer, LFW Dataset, Deep Learning**

I. Introduction

Face recognition has become pivotal in security, authentication, and humancomputer interaction. Convolutional neural networks (CNNs) traditionally dominated this field, but transformer-based approaches have shown promising results by modeling global dependencies within images [2]. This work investigates and compares ViT and Swin Transformer architectures on the challenging LFW dataset [1], which contains over 13,000 images of faces in the wild.

II. Related Work

A. LFW Dataset

The LFW dataset [1] is a benchmark for unconstrained face recognition, featuring real-world variations in pose, lighting, and occlusion. It has been extensively used to evaluate advances in face verification algorithms [9].

B. Vision Transformer

The Vision Transformer (ViT) [2] adapts the transformer architecture to image patches, demonstrating competitive performance on large-scale datasets. It splits an input image into fixed-size patches, embeds them, and processes them via multi-head self-attention.

C. Swin Transformer

The Swin Transformer [3] introduces a hierarchical, shift-windowing scheme to efficiently capture local and global context. It achieves state-of-the-art performance on multiple vision tasks while maintaining lower computational overhead.

III. Dataset

We utilize the LFW dataset, accessible via Kaggle [6] and officially described in [1]. It comprises 13,233 images across 5,749 identities. We adopt an 8020 train-test split, ensuring identity-disjoint sets.

IV. Methodology

A. Data Preprocessing

Images resized to 224×224, normalized using ImageNet statistics. Training augmentations include random horizontal flip and color jitter.

```
from torchvision import transforms, datasets
transform_train = transforms.Compose([
    transforms.Resize((224,224)),
    transforms.RandomHorizontalFlip(),
    transforms.ColorJitter(),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485,0.456,0.406],std=[0.229,0.224,0.225])
])
dataset_train = datasets.ImageFolder('data/lfw/train', transform=transform_train)
loader_train = torch.utils.data.DataLoader(dataset_train, batch_size=64, shuffle=True)
```

Listing 1. DataLoader and Transforms

B. Model Architectures

1) Vision Transformer:We use the Hugging Face implementation of ViT-Base [4], fine-tuned for 50 epochs.

```
from transformers import ViTForImageClassification, ViTConfig
config = ViTConfig.from_pretrained('google/vit-base-patch16-224')
model_vit = ViTForImageClassification.from_pretrained('google/vit-base-patch16-224', config=config)
```

Listing 2. ViT Model Initialization

2) Swin Transformer: We employ the Swin-B variant via Hugging Face [5], fine-tuned for 30 epochs.

```
from transformers import SwinForImageClassification, SwinConfig
config = SwinConfig.from_pretrained('microsoft/swin-base-patch4-window7-224')
model_swin = SwinForImageClassification.from_pretrained('microsoft/swin-base-patch4-window7-224', config=config)
```

Listing 3. Swin Transformer Initialization

C. Training Loop

Both models trained with AdamW optimizer [7] and cosine learning rate schedule.

```
from torch.optim import AdamW
from transformers import get_cosine_schedule_with_warmup
optimizer = AdamW(model.parameters(), lr=3e-4)
scheduler = get_cosine_schedule_with_warmup(
    optimizer, num_warmup_steps=500, num_training_steps=total_steps
)
for epoch in range(epochs):
    model.train()
    for batch in loader_train:
```

```
        inputs, labels = batch
        outputs = model(**inputs)
        loss = outputs.loss
        loss.backward()
        optimizer.step(); scheduler.step();
    optimizer.zero_grad()
```

Listing 4. Training Loop Snippet

V. Experimental Setup

All experiments conducted on a single NVIDIA RTX 3090 Ti GPU, using PyTorch 1.13 and Transformers 4.28. Metrics include top-1 accuracy and ROC AUC via scikit-learn [8].

VI. Results and Discussion

Table I compares performance.

TABLE I
Performance on LFW Dataset

Model	Accuracy (%)	AUC	Inference Time (ms)
ViT-Base	95.8	0.971	18
Swin-B	99.2	0.998	12

Swin Transformer outperforms ViT in accuracy and AUC, and is 33% faster during inference due to hierarchical windowing.

VII. Conclusion

This study evaluated the performance of Vision Transformer (ViT) and Swin Transformer models for face recognition using the Labeled Faces in the Wild (LFW) dataset. By fine-tuning pretrained versions of both models, we observed that the Swin Transformer significantly outperformed ViT in terms of accuracy (99.2% vs. 95.8%), ROC AUC (0.998 vs. 0.971), and inference time (12ms vs. 18ms). These results highlight the effectiveness of Swins hierarchical architecture and localized self-attention mechanism, which allow for better feature extraction and generalization, especially on relatively small datasets like LFW.

In contrast, ViTs patch-based self-attention, while powerful, was less effective without large-scale training data. Our findings suggest that Swin Transformer is a more practical and accurate choice for face recognition in real-world settings. Future work can expand on this by incorporating more transformer variants, applying domain-specific pretraining, or optimizing models for edge deployment. Overall, the Swin Transformer stands out as a robust solution for accurate and efficient face recognition tasks.

Acknowledgment

We would like to express our sincere gratitude to our instructor, Ms. Sumaiyah Zahid, for their continuous support, valuable guidance, and insightful feedback throughout this project. Their encouragement and mentorship played a pivotal role in the successful completion of our work.
We also acknowledge the contributions of the open-source community and researchers whose tools and datasets, particularly the Labeled Faces in the Wild (LFW) dataset, were instrumental in our experimentation.
Finally, we are thankful to the FAST computing cluster support team.

References

[1] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, University of Massachusetts, Amherst, Tech. Rep., 2008.
[2] A. Dosovitskiy *et al.*, An Image is Worth 16Œ16 Words: Transformers for Image Recognition at Scale, arXiv:2010.11929, 2020.
[3] Z. Liu *et al.*, Swin Transformer: Hierarchical Vision Transformer using Shifted Windows, in Proc. ICCV, 2021.
[4] Hugging Face, ViT: Vision Transformer, https://huggingface.co/docs/transformers/en/model_doc/vit, accessed Apr. 2025.
[5] Hugging Face, Swin Transformer, https://huggingface.co/docs/transformers/model_doc/swin, accessed Apr. 2025.
[6] J. Li, LFW Dataset, Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/jessicali9530/lfw-dataset
[7] I. Loshchilov and F. Hutter, Decoupled Weight Decay Regularization, in ICLR, 2019.
[8] F. Pedregosa *et al.*, Scikit-learn: Machine Learning in Python, JMLR, vol. 12, pp. 28252830, 2011.
[9] E. Learned-Miller *et al.*, LFW Results and Protocol, https://people.cs.umass.edu/~elm/papers/lfw.pdf, accessed Apr. 2025.