Supervised ML Project: Classification of Skin Cancer

BMED 6517

Group 4

Celine Al-Noubani, Soobin An, Sharon Kartika, Daniel Lai

7/21/2025

Tables of Contents

- Introduction
- Dataset & Preprocessing
- Method Overview (Pipeline)
- Baseline Model: CNN
- Final Model: ViT
- Results: CNN vs. ViT
- Discussion & Conclusion

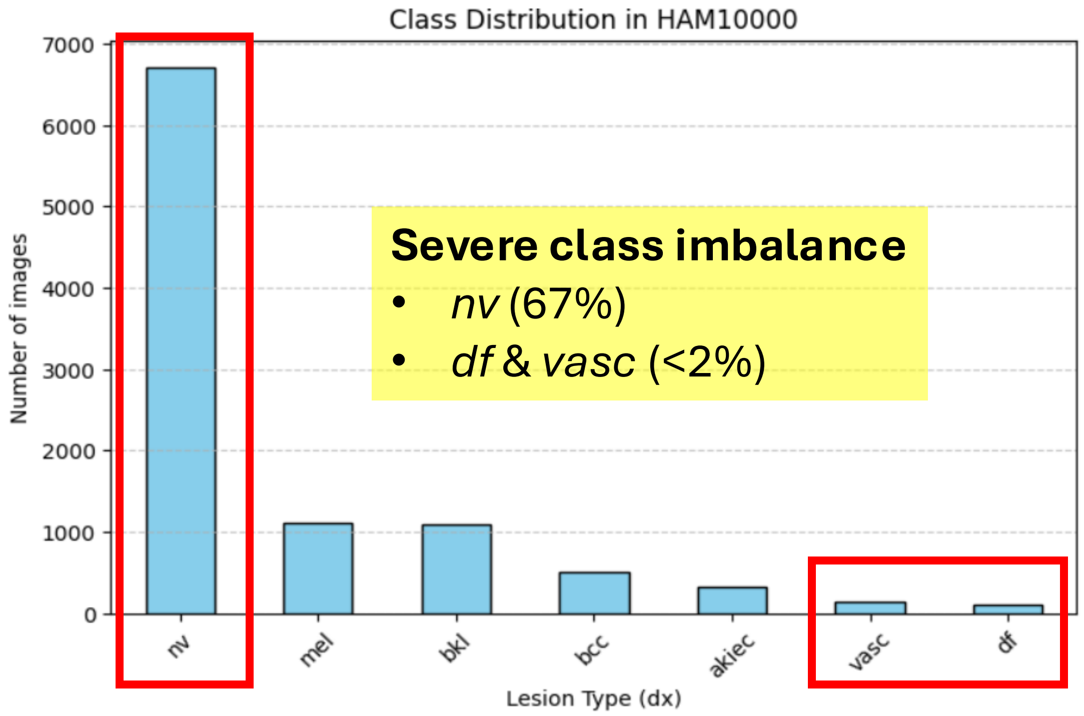
Why Skin Lesion Classification?

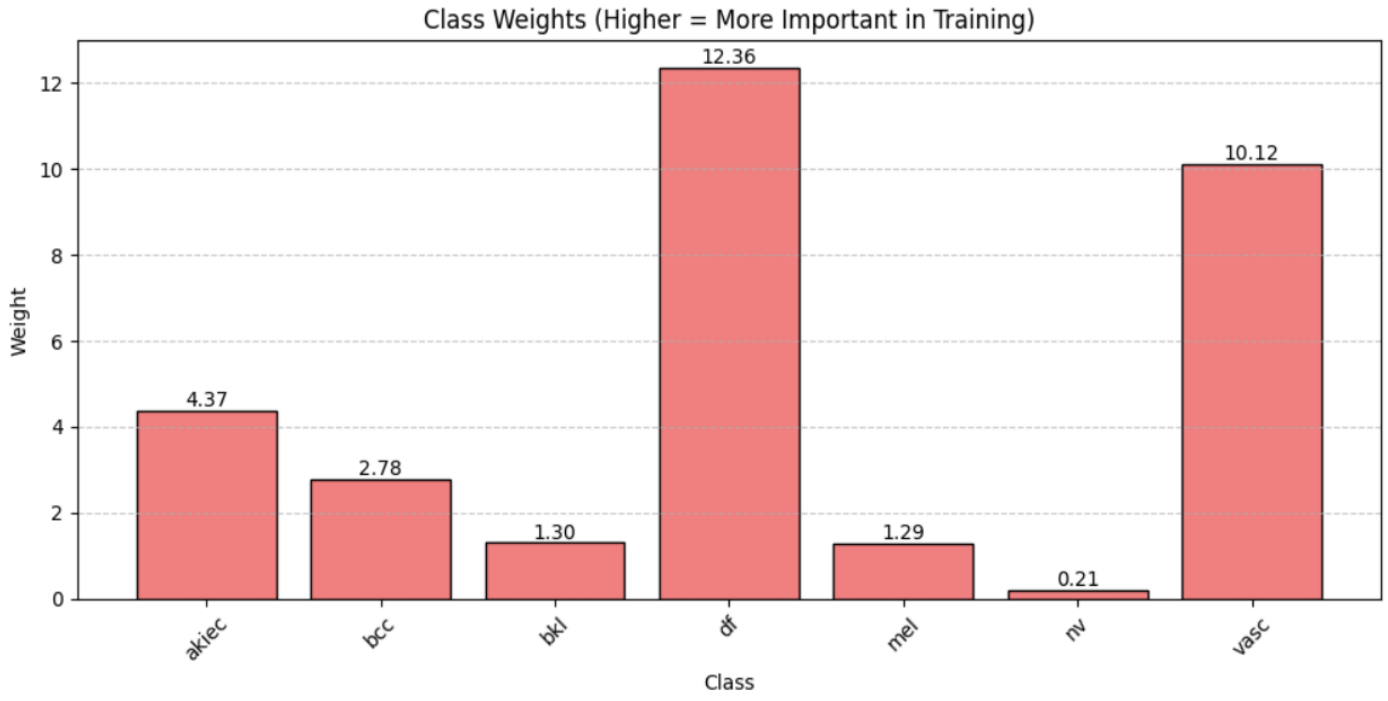
- Skin cancer is one of the most common cancers worldwide[1]
  - Early detection is critical.
- Dermoscopy helps non-invasive diagnosis but is subjective[1]
- Automated image-based systems are needed[2-4]
  - CNNs: Good at local features but struggle with global context & imbalance
  - ViTs: Captures global patterns but need large, balanced data
- Goal: Compare CNN vs ViT on small, imbalanced dataset (HAM10000)

HAM10000 Dataset & Preprocessing

- 10,015 dermoscopic images
- 7 classes: ***akiec***, ***bcc***, ***bkl***, ***df***, ***mel***, ***nv***, and ***vasc***
- Severe class imbalance
  - *nv* (67%)
  - *df* & *vasc* (<2%)

Class Imbalance Handling: CNN vs. ViT





CNN

- Applied class weights → penalize errors in minority classes more

ViT

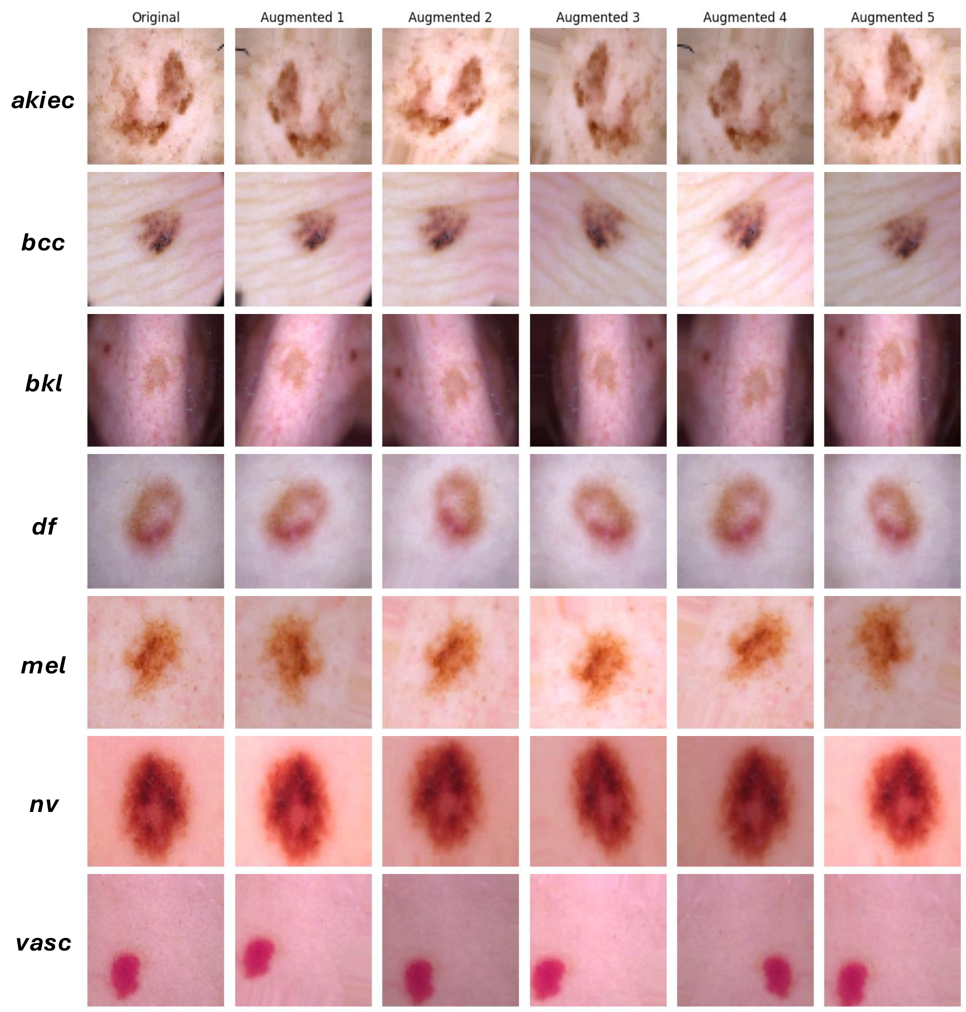
- Applied oversampling → minority classes **duplicated** to balance the training set

Data Augmentation: CNN vs. ViT

Purpose:

Data diversity ↑ & Overfitting ↓

Technique	CNN (Keras)	ViT (PyTorch)
Input Size	64×64×3	224×224
Rotation	±20°	±50°
Translation	Width/Height ±10%	RandomAffine ±10%
Zoom	±10%	±10%
Flipping	Horizontal	Horizontal
Brightness	0.8–1.2	0.8–1.2
Normalization	[0,1] scaling	[0,1] scaling



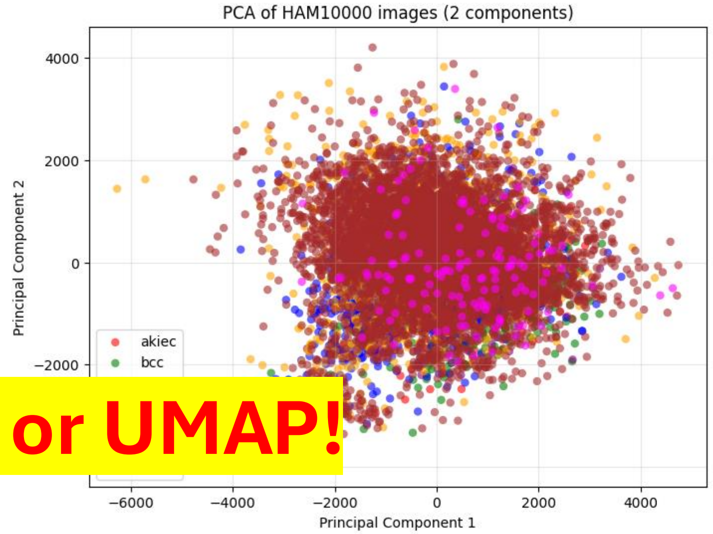
Preprocessing Result

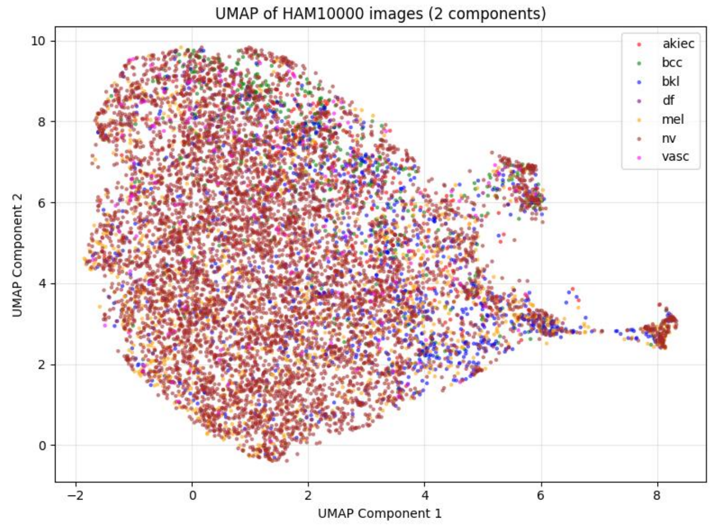
Purpose:

Data diversity ↑ & Overfitting ↓

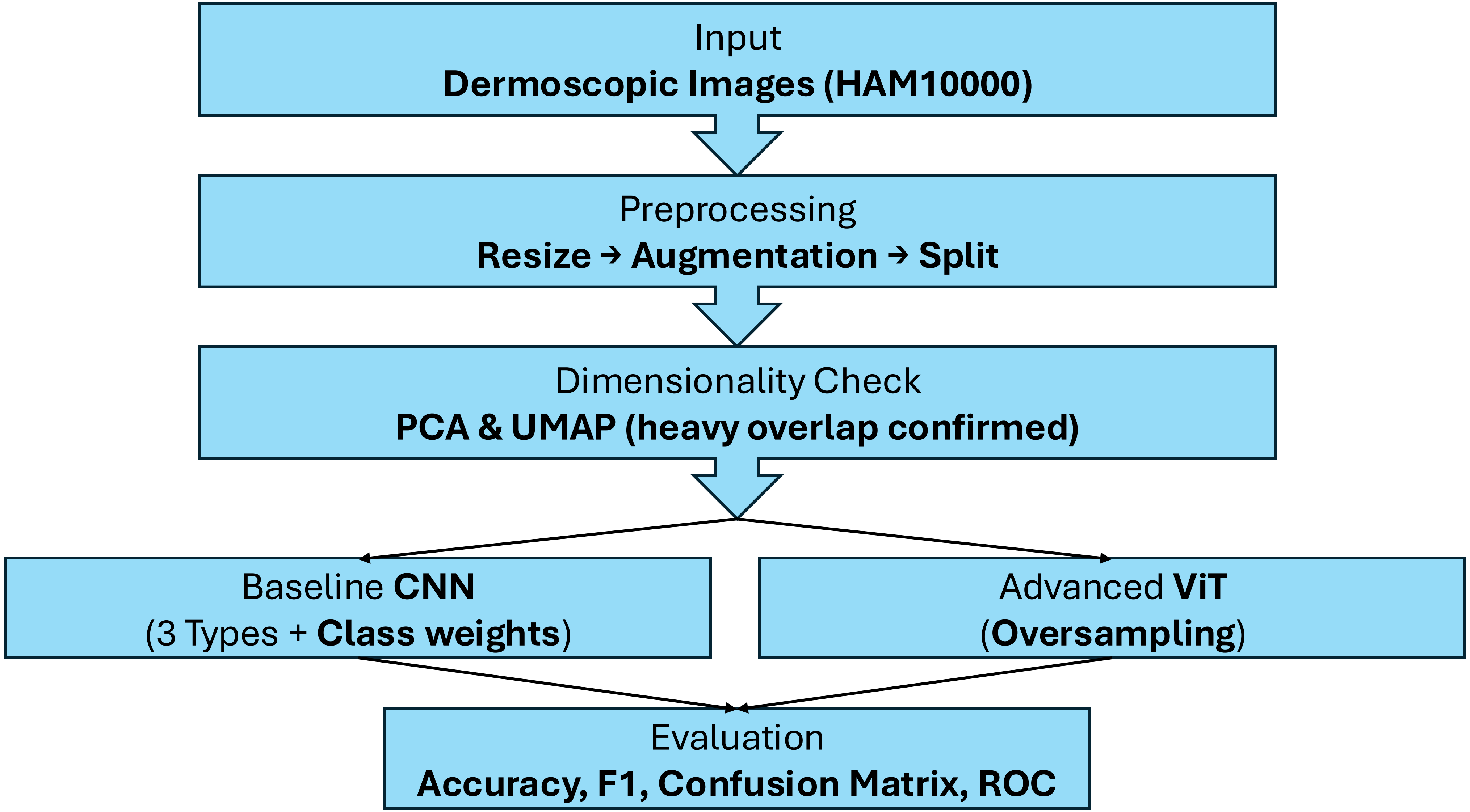
Technique	CNN ([illegible]	[illegible]
Input Size	64×64 [illegible]	[illegible]
Rotation	±20°	±50°
Translation	Width/Height ±10%	RandomAffine ±10%
Zoom	±10%	±10%
Flipping	Horizontal	Horizontal
Brightness	0.8–1.2	0.8–1.2
Normalization	[0,1] scaling	ImageNet stats

Not Easy to Classify with PCA or UMAP!





Modeling Pipeline Overview



CNN Model Progression

Model	Key Features / Changes	Validation Accuracy
Baseline	- 2 Conv blocks (32, 64) - Dense(128), Dropout(0.5) - LR = 0.001, Class Weights - 10 epochs + EarlyStopping - Cross-entropy loss	~58%
Enhanced CNN	- Added 3rd Conv block - Doubled filters: (32, 64, 128) - Dense(256), Dropout(0.4) - BatchNorm added - LR ↓ to 0.0005, ReduceLROnPlateau - 30 epochs	~64%
Focal Loss	- Focal loss (γ=2.0) - LR ↓ to 0.0001 - Dropout(0.3) - Epochs = 25	**~76%**

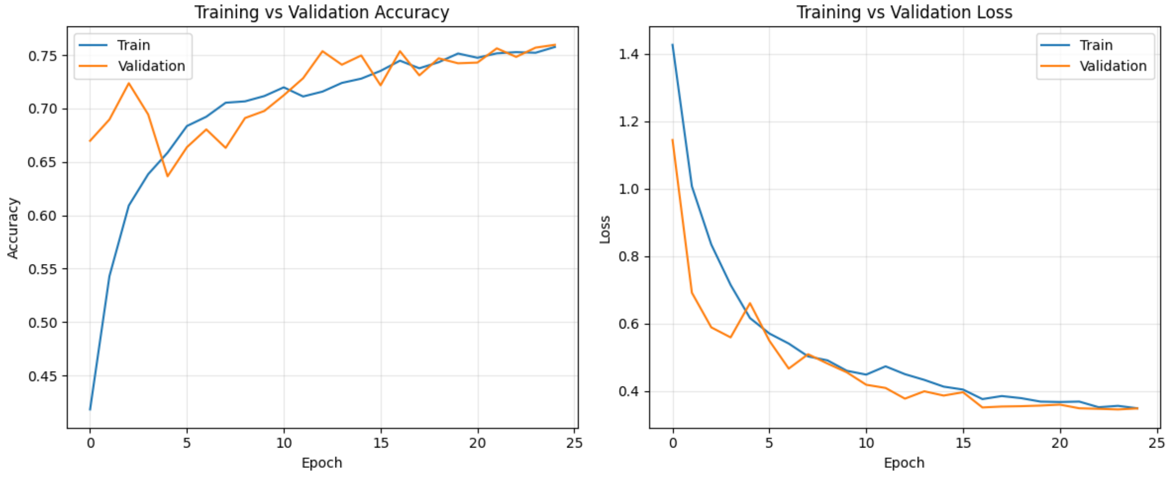
CNN Per Model Metrics Summary

Model	Test Accuracy	Test Loss
Basic CNN	53.96%	1.1721
Enhanced CNN	62.61%	0.9148
Focal Loss CNN	76.18%	0.3431

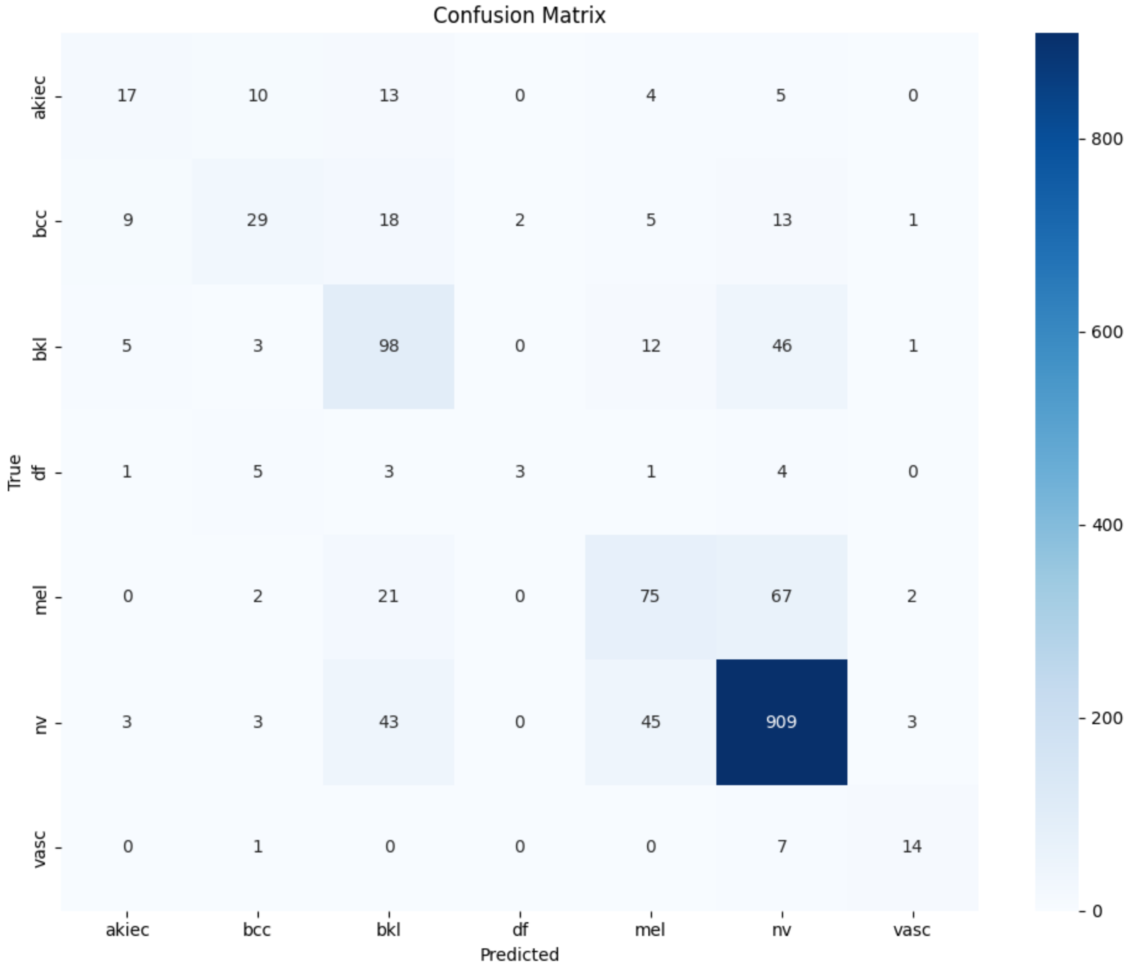
CNN Final Model Classification Report

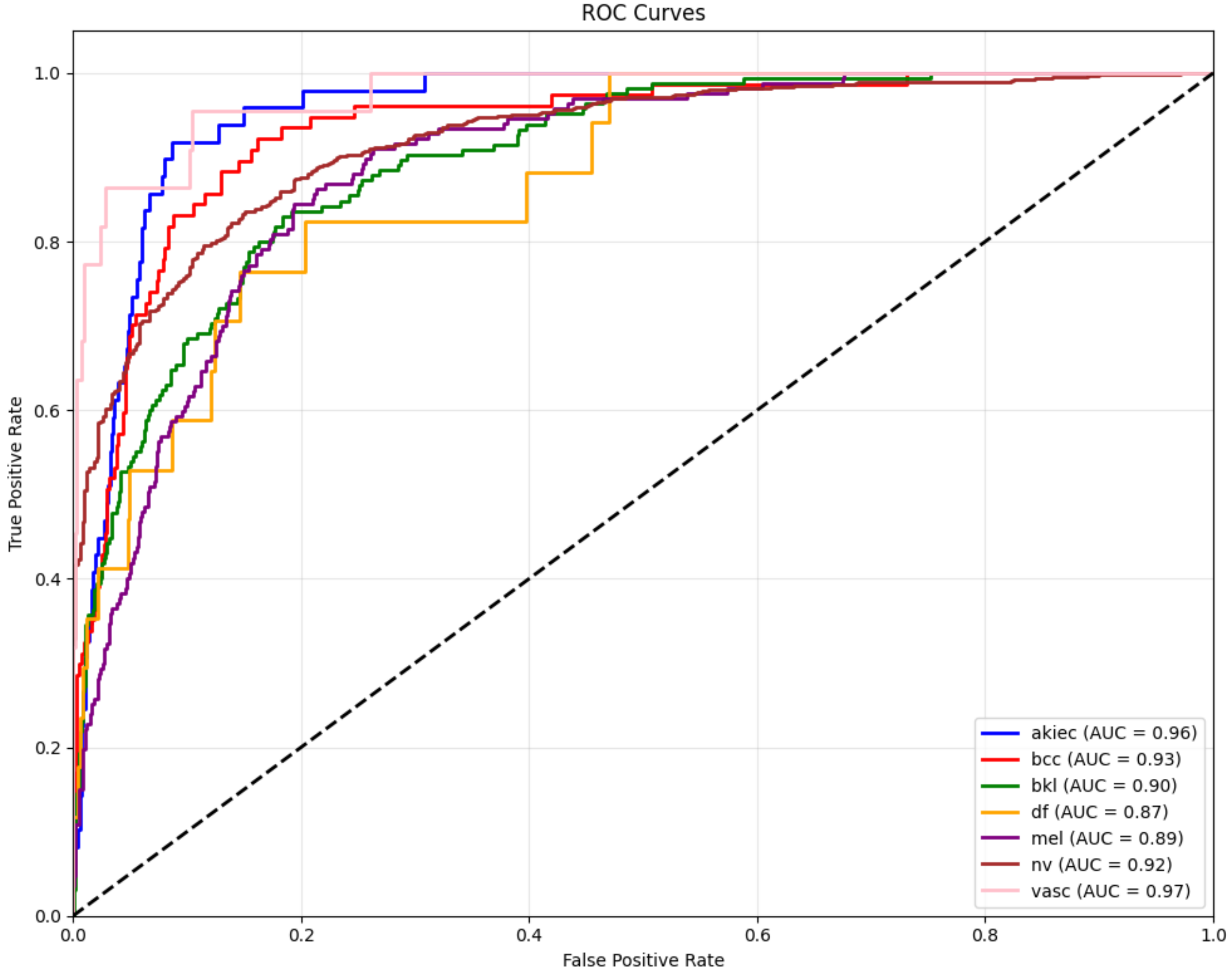
Classification Report:

	precision	recall	f1-score	support
akiec	0.49	0.35	0.40	49
bcc	0.55	0.38	0.45	77
bkl	0.50	0.59	0.54	165
df	0.60	0.18	0.27	17
mel	0.53	0.45	0.49	167
nv	0.86	0.90	0.88	1006
vasc	0.67	0.64	0.65	22
accuracy			0.76	1503
macro avg	0.60	0.50	0.53	1503
weighted avg	0.75	0.76	0.75	1503



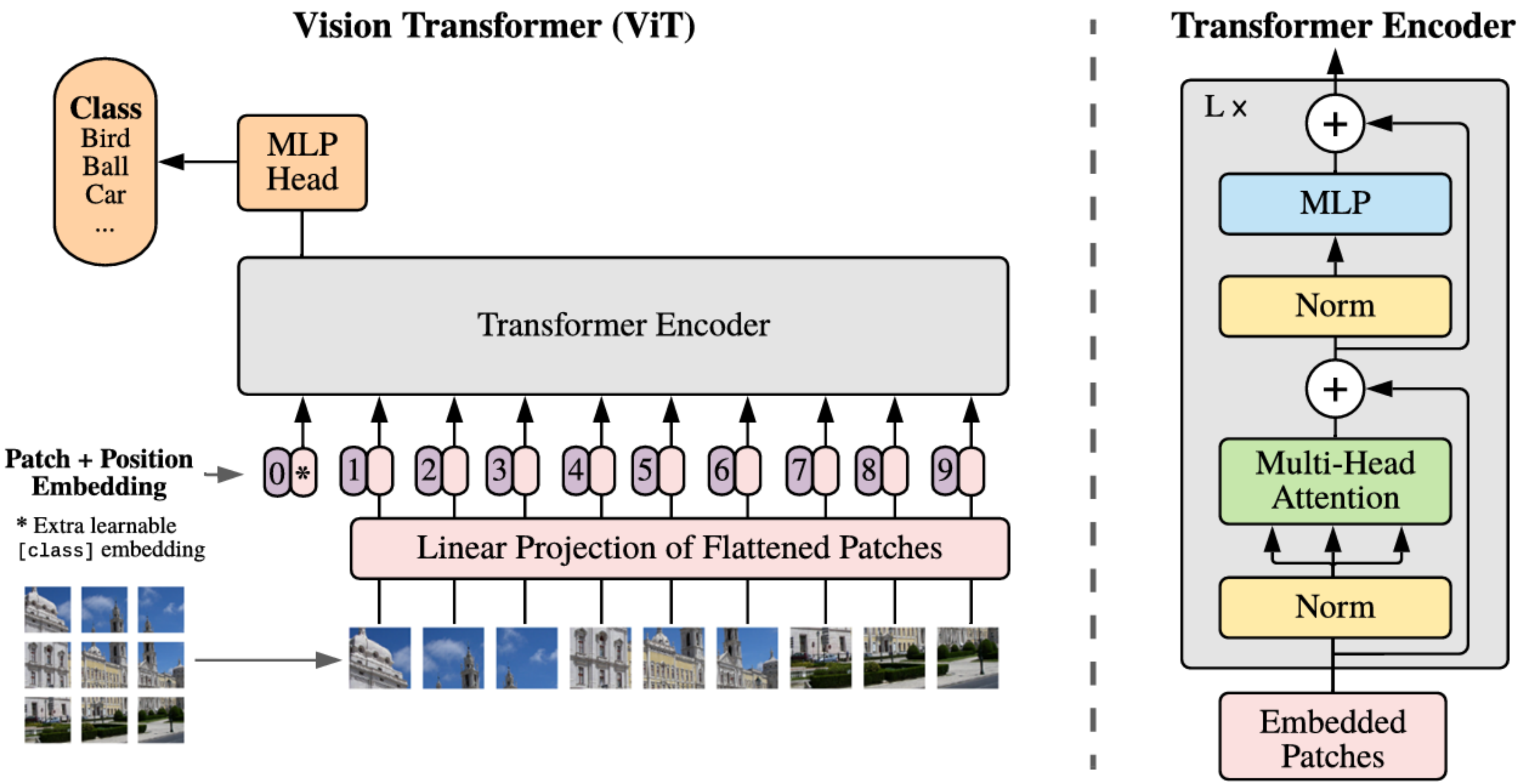
CNN Final Model Confusion Matrix & ROC





ViT Architecture

Divide into patches
Linearly embed
Add position embeddings
Add classification token
Feed to a transformer encoder



ViT Model

`google/vit-base-patch16-224-in21k`

Trained on ImageNet-21k (14 million images, 21,843 classes)

Fine-tuned on HAM10000

12 Transformer encoder layers

~ 86 million parameters

```
ViTForImageClassification(
  (vit): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-11): 12 x ViTLayer(
          (attention): ViTAttention(
            (attention): ViTSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (layernorm_after): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
    (layernorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
  )
  (classifier): Linear(in_features=768, out_features=7, bias=True)
)
```

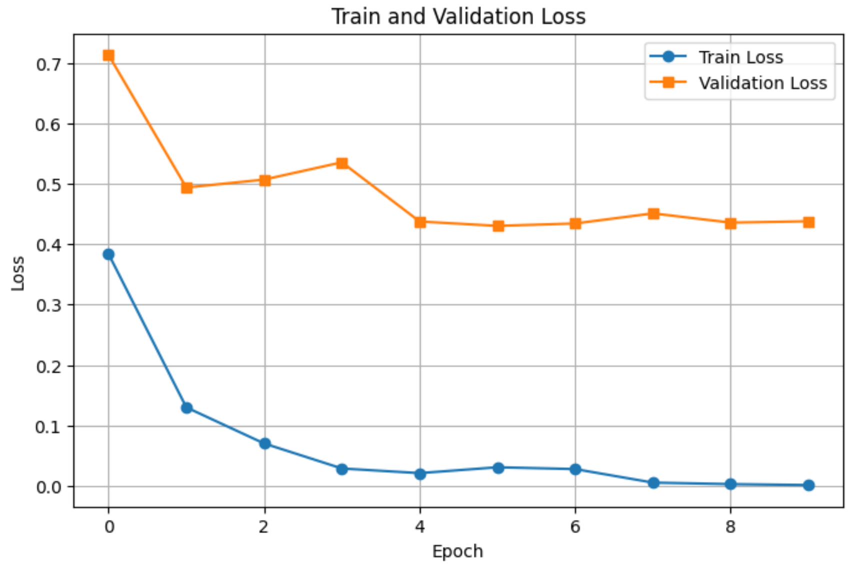
ViT Training

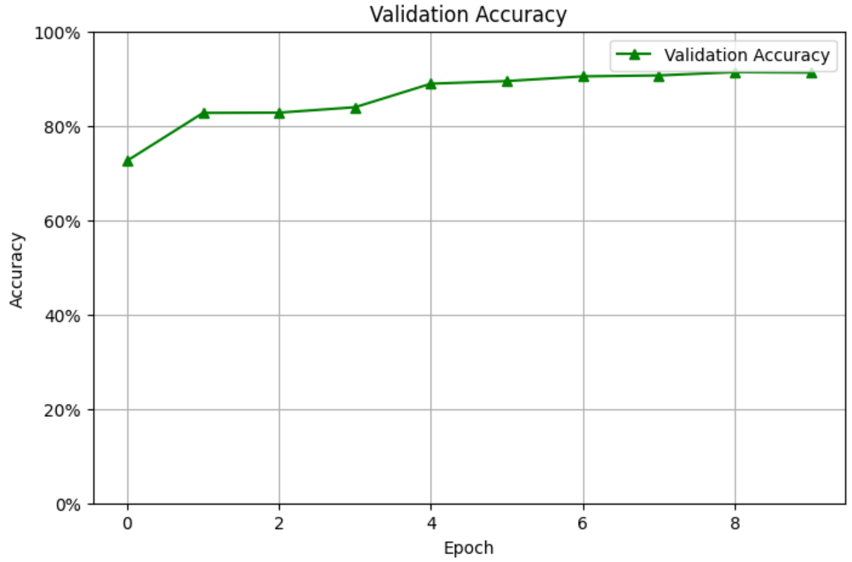
10 epochs on 80% of the training data

Number of training examples after oversampling:
5352 per class x 7 classes = 37464

Took 2 hours on M2 ultra

	precision	recall	f1-score	support
akiec	0.95	0.95	0.95	21
bcc	0.81	0.76	0.79	198
bkl	0.85	0.84	0.84	219
df	0.95	0.97	0.96	1353
mel	0.72	0.62	0.67	61
nv	0.89	0.87	0.88	126
vasc	0.85	0.92	0.88	25
accuracy			0.91	2003
macro avg	0.86	0.85	0.85	2003
weighted avg	0.91	0.91	0.91	2003

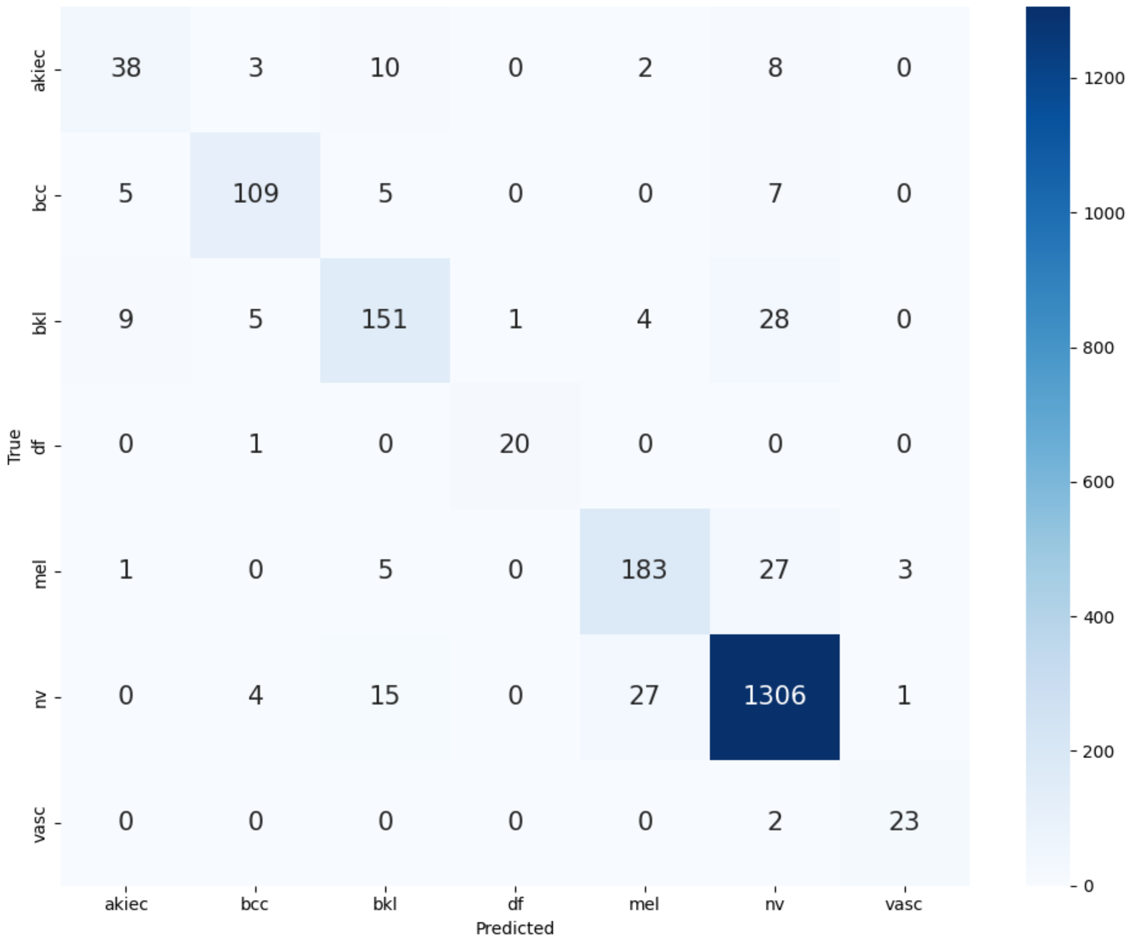


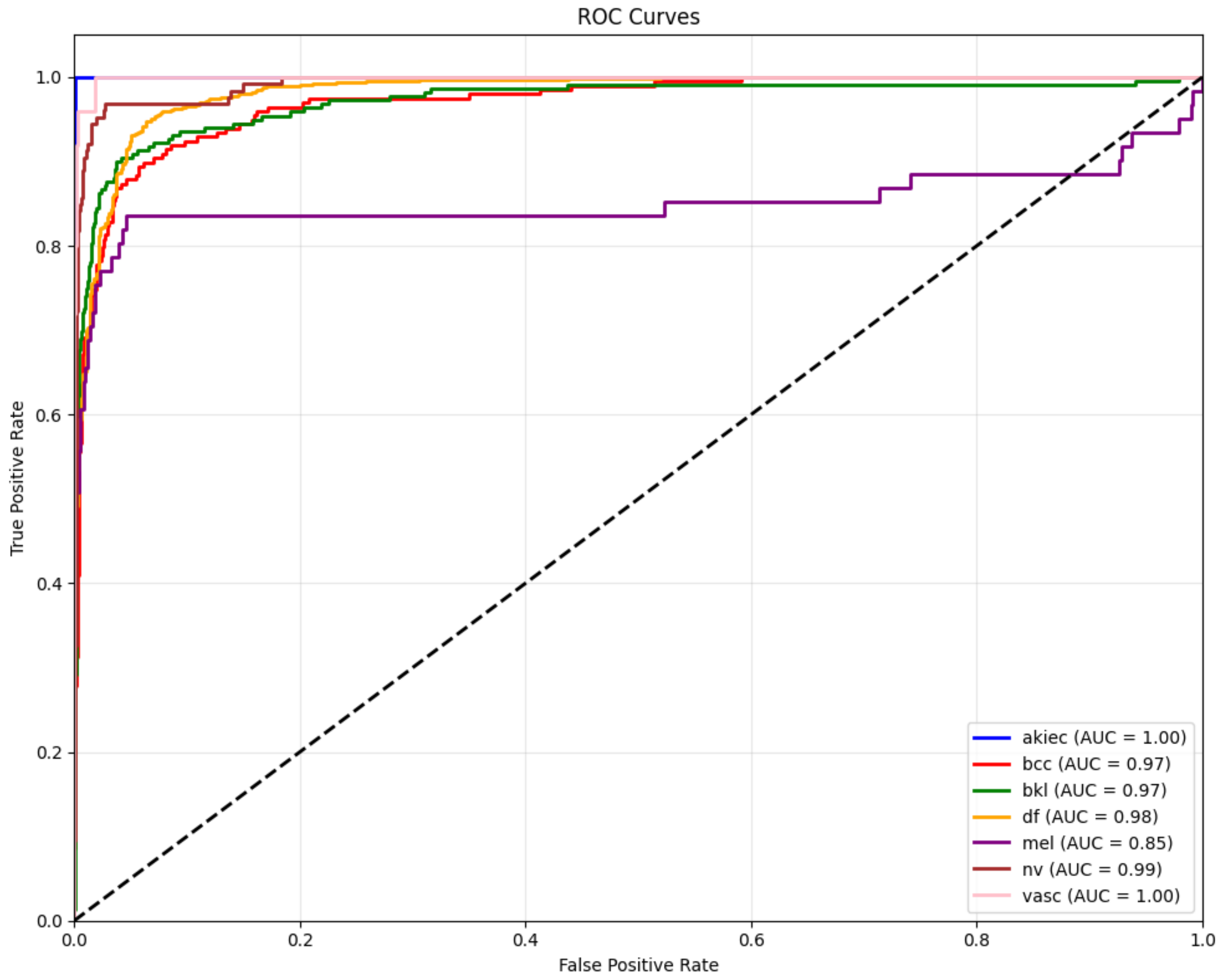


ViT Final Model

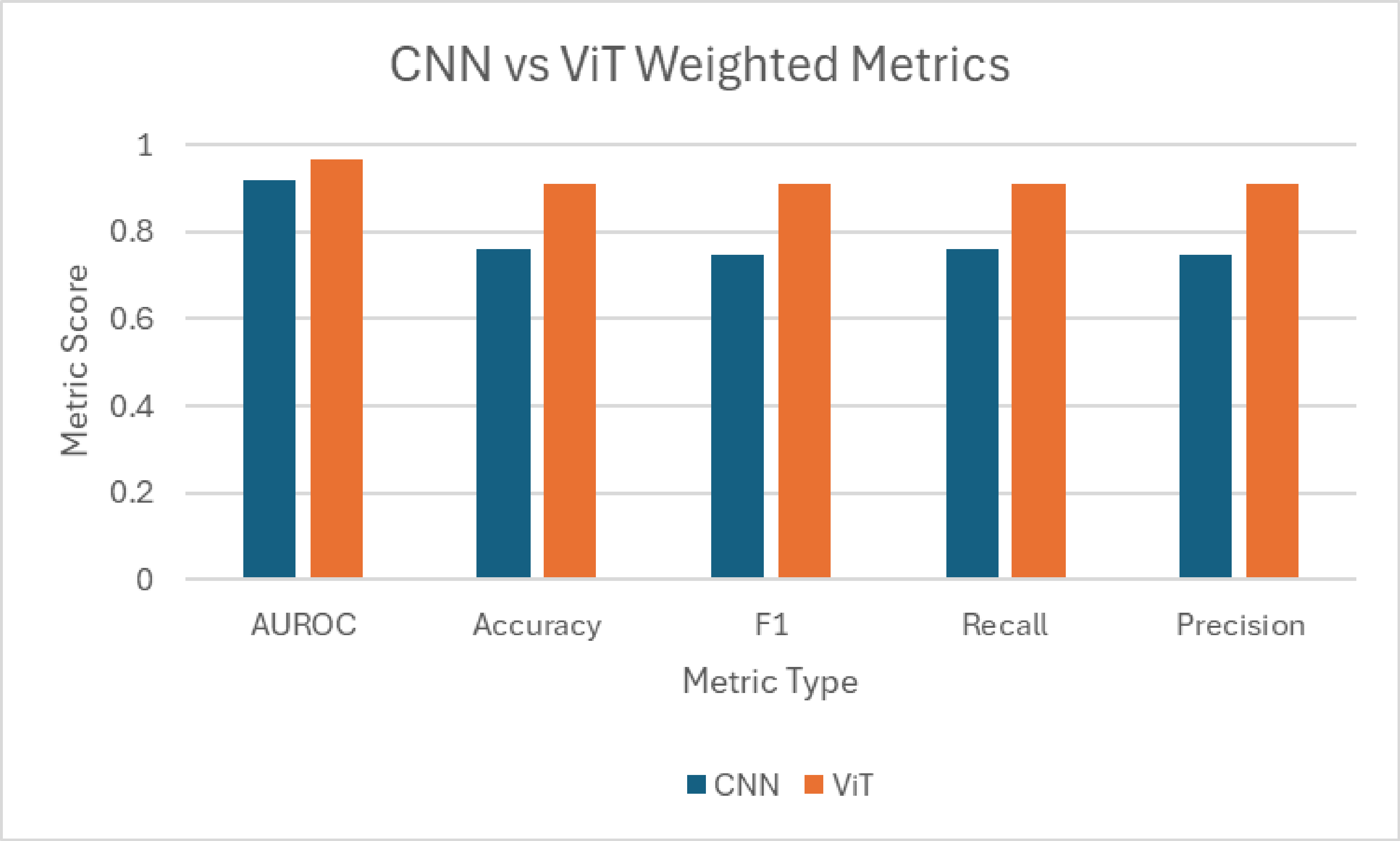
- Total accuracy jumps from 76% to 91%
- Lowest precision is 72% on mel (49% on akiec with CNN)

ViT Confusion Matrix & ROC

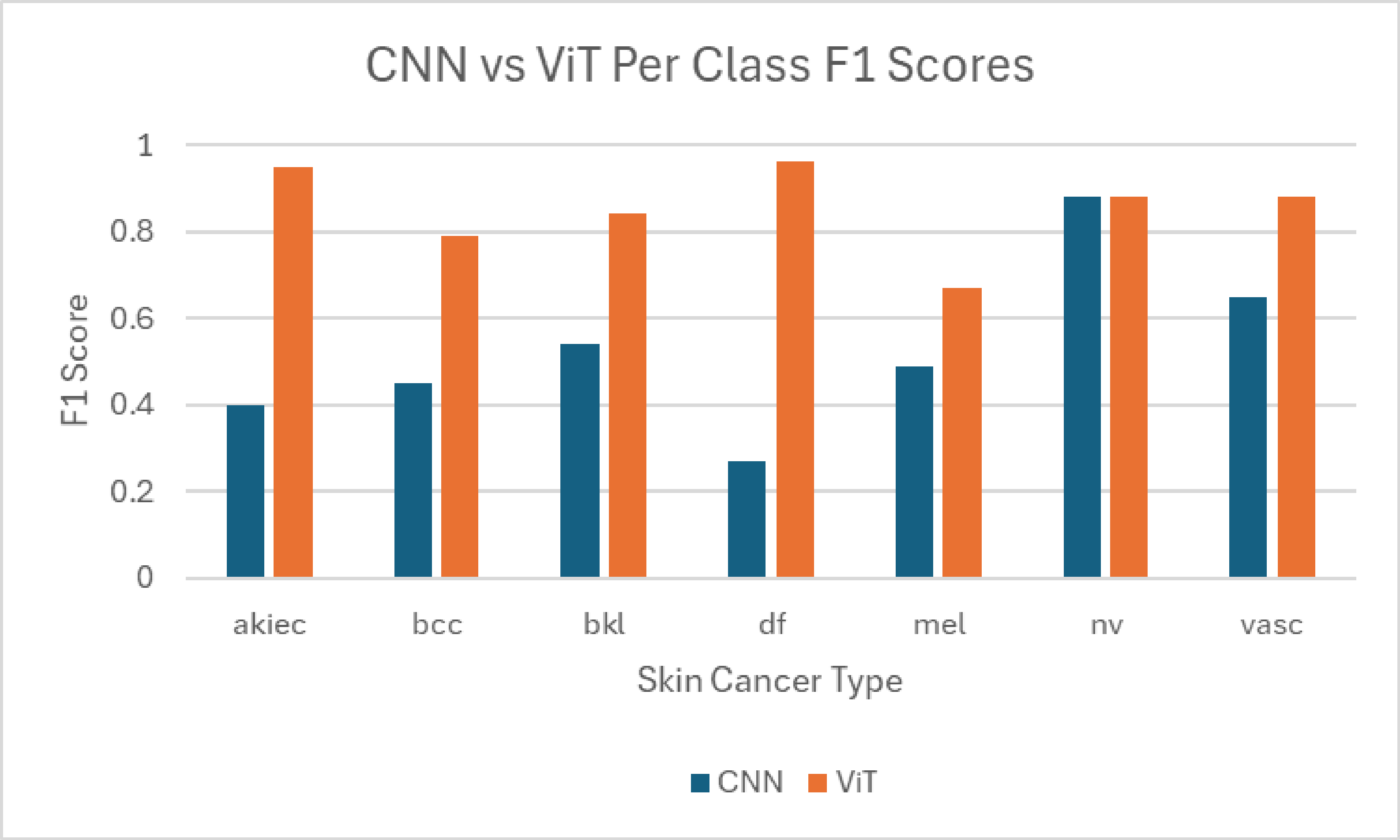




CNN vs ViT: Weighted Metrics



CNN vs ViT: Per Class F1



Discussion

Strengths/weakness

CNN

- Strong performance on imbalanced, small datasets; effective with localized features
- Limited global context capture

ViT

- Good with long-range dependencies; high performance on dominant classes
- Struggles with minority class detection without proper balancing

Conclusion + Future Work

Key takeaways

- CNN: *“Trade-off”*: **Focal Loss** boosted hard sample learning but reduced minority class sensitivity; lower accuracy, but higher consistency
- ViT: Showed general higher accuracy, but difficulties with minority classes, but improves significantly on oversampling

Future work

- CNN: Improved optimization of hyper parameters; could pre-train CNN with different dataset
- ViT: Investigate better global pattern recognition & minority class handling

References

1. Codella, N., Rotemberg, V., Tschandl, P., Celebi, M. E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., Kittler, H., & Halpern, A. (2018). *Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)*. arXiv. https://arxiv.org/abs/1902.03368
2. Maurício, J., Domingues, I., & Bernardino, J. (2023). Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. *Applied Sciences*, *13*(9), 5521. https://doi.org/10.3390/app13095521
3. Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data, 5*, 180161. https://doi.org/10.1038/sdata.2018.161
4. Mader, K. (2018). *Skin Cancer MNIST: HAM10000*. Kaggle. https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000
5. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021)., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. (arXiv:2010.11929v2). arXiv. https://doi.org/10.48550/arXiv.2010.11929