ViT vs. CNN on Elephants

Ariel Lulinsky[a*], Hadar Hai[b]
[a]Dept. of Electrical Engineering, Technion, Haifa, Israel
[b]Technion Autonomous Systems Program (TASP), Technion, Haifa, Israel

Abstract

This project develops and compares Convolutional Neural Networks (CNN) and Vision Transformers (ViT) for an image classification task of Asian and African elephants. It shows that ViT outperforms CNN when using a transfer learning technique utilizing state-of-the-art pretrained model on large datasets.

Keywords: CNN; ViT; Transfer learning; Supervised; Self-supervised

1 Overview

This project aims to develop and compare Convolutional Neural Networks (CNN) and vision Transformers (ViT) for an image classification model to distinguish between Asian elephants and African Elephants. By using a transfer learning technique utilizing state-of-the-art pretrained models we can analyze and classify images of these majestic creatures with high accuracy. This project uses a publicly available dataset and aims to help elephant conservation and research.

2 Introduction

Over the last years, there has been a shift to utilize a new architecture of transformers for various Natural Language Processing (NLP) tasks. It showed great improvement and outperformed previous methods such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) . Following the success in NLP, it was suggested that with some modifications, it might be useful also for Computer Vision tasks. Previous works showed that it can outperform traditional CNNs for image classification tasks [1]. Therefore, we looked for a publicly available binary classification task that hasn't yet been widely explored and has some extent of difficulty. Our main goal was to endorse the hypothesis that ViT outperforms CNN for binary classification tasks. Besides, we also wanted to analyze Low Rank Adaption Fine-Tuning (LoRA) and Weight-Decomposed Low Rank Adaptation (DoRA). Moreover, we wanted to check the performance of supervised versus self-supervised pretrained ViTs.

3 Ethics

- **Stakeholders**: safari employees, nature researchers, zoologists, conservationists, wildlife organizations.
- **Implications**: Safari employees may save time and effort in distinguishing two species for proper treatment but may also lose their expertise. Zoologists may have better control over the survival of elephants in Africa and Asia but may experience a lack of physical and emotional connection to them. High performance of our method may help conservationists in their efforts to monitor and protect these endangered species. However, low performance may lead to an imbalance in wildlife ambiance and extinction danger.
- **Ethical Considerations**: Addressing biases in the dataset to prevent discriminatory outcomes. Ensuring transparency and explainability in how classification is deployed. Encouraging collaboration between AI and zoologists rather than replacement. Promoting awareness and understanding of the distinct characteristics and conservation needs of Asian and African elephants.

4 Method

In this section, we will discuss our algorithm, training, architecture, and hyperparameters.

- **Algorithm** - we used transfer learning technique for state-of-the-art pretrained CNN and ViT models. The pretrained models were trained on ImageNet - a huge dataset containing 14 Million images classified into 20,000 classes.

  For CNN, we chose to work with Mobile Net that showed the best results for this task [2] and Resnet that is a popular architecture [5].

  For ViT, we chose one supervised model and one self-supervised model. At first, we modified the last layer: Fully Connected (FC) to be with an output of two classes. Later on, for CNN, we also suggested replacing another FC layer with a lot of parameters by LoRA/DoRA.

- **Training** - We initialized our weights with the pretrained weights for all the layers except the last one, which we initialized randomly.

  We had two options:

  [1] - Feature Extraction (FE): Freeze the weights for all the layers except the last one and train only on the last one layer. This saves a lot of time.

  [2] - Fine Tuning (FT): Train the weights for all layers. This takes more time but might yield better results if the previous method's results were not sufficient.

  The train was done with cross-entropy loss.

- **Architecture** - Since we didn't build an architecture from scratch, we just cover the main concepts of the architectures we worked with.

  The CNN architectures include blocks of convolutions, where a typical block composed of a convolutional layer followed by batch norm, activation function, and pooling. The last block contains FC layers and a softmax to get an output of classes' probabilities. This architecture looks for patterns of shapes and colors that distinguish between image classes most consistently.

  The supervised ViT architecture splits each image into a sequence of fixed-size patches (16x16 in our case), which are linearly embedded. Each embedded patch becomes a token and the resulting sequence of embedded patches with a positional embedding passes to the model. The model itself is a typical transformer encoder and the output projected into desired classes using an FC head [1].

  This architecture looks for relationships between different parts of images and creates relational information. The ViT architecture is shown in Figure 1.

  The self-supervised ViT architecture is DINOv2. It is an improved version of DINOv2, which uses two ViT networks - student and teacher and compares their output with respect to two different augmentations for the same input image. Then teacher parameters are updated with an Exponential Moving Average (EMA) of student parameters [6].

- **Hyperparameters** - We suggested a set of parameters that was used in previous works and fine-tuned them using a validation set to find the best for our task. Our hyperparameters were:

  1. Batch size: {32, 64, 128}
  2. Epochs: {15, 20, 25}
  3. FE/FT
  4. LoRA/DoRA/same FC

  We found the best parameters to be batch size = 32, epochs = 20, FE, and the same FC.

  Other parameters we used without tuning:

  1. Optimizer: Stochastic Gradient Descent (SGD)
  2. Learning rate: 0.001
  3. Momentum: 0.9

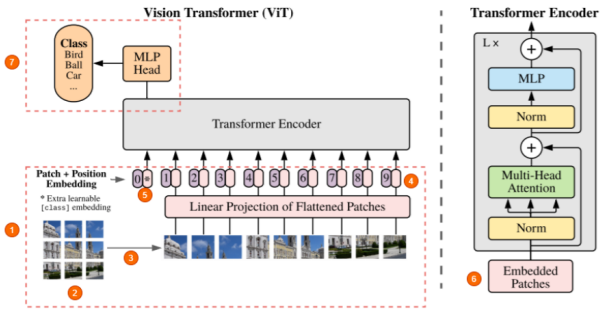

Figure 1: Vision Transformer Architecture

5 Experiments and Result

5.1 Dataset

Repository location - Asian vs African Elephants (kaggle.com)

The dataset contains 1028 images classified into two classes - African and Asian elephants. It contains 840 images for the train set and 188 images for the test set. The train set is uniformly divided over classes, with 420 images for each class. The test set contains 97 images of African elephants and 91 of Asian ones. We separated the train set into train and validation sets with a ratio of 0.1% . The summarized information is given in Table 1.

	African	Asian	% of Total
Train	369	369	70
Validation	51	51	10
Test	97	91	20

Table 1: Data distribution

Some data characteristics:

- Different image shapes - ranges from (100, 100) to (4992, 3328).
- To increase complexity train set contains less than 5% mislabeled images.
- All images in the test set have the correct label (apparently - it proved wrong as we looked deeper on the test set).
- Balanced distribution for the train and test sets.

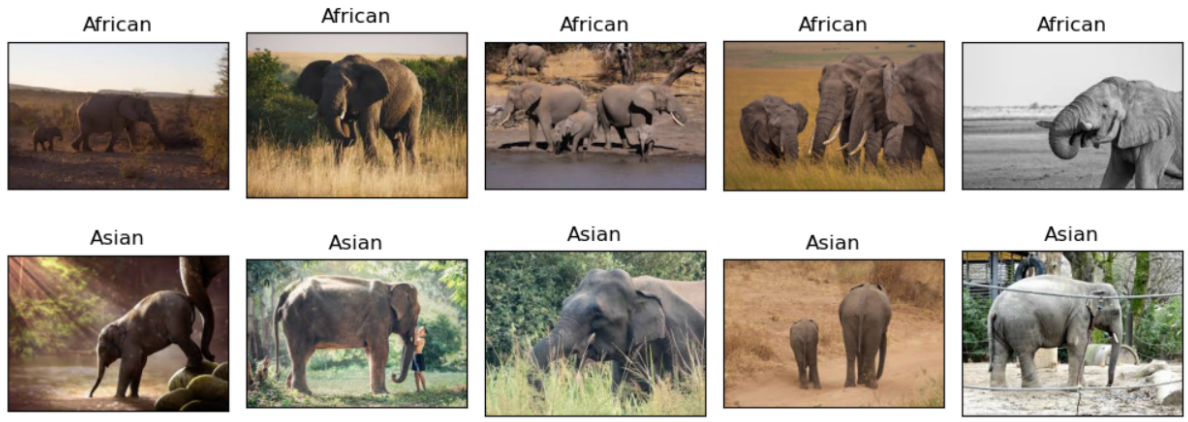

Figure 2: Data Examples

5.2 Workflow

Our workflow contains the following steps:

1. Creation of validation set - by splitting training set such that we will have 70% train, 10% validation, and 20% test sets.
2. Data augmentations - to improve results, avoid overfitting and helping the network generalize better we applied the following augmentations using Kornia on the train set: cropping, scaling, rotation, horizontal flip, and zoom. For validation and test set we only applied resizing to the fixed input size - (224, 224) and we normalize all the images by the parameters of ImageNet training set.
3. Fine-tuning - we examined several sets of parameters (batch size and epochs) for Mobile Net architecture and found the best one using the validation set. Then we applied LoRA, DoRA, and FT to the best model but it hadn't improved the results.
4. Train the different models - using the best parameters we have found, we trained the four architectures - Mobile Net, Resnet, ViT-b-16, and DINOv2.

5. Comparing results - we analyzed the best CNN model and the best ViT model. The comparison included test accuracy, confusion matrix, vision results on test images, incorrect classifications, and activation maps (unfortunately, we were unable to plot ViT attention maps but we did it for CNN).

5.3 Results

We didn't get improvement using LoRA or DoRA for Mobile Net probably because it replaced only one linear layer with not so much parameters (576X1024) comparing to VGG16 discussed in tutorial 9. In addition, we didn't get an improvement using FT probably because we had a very small dataset. We would also like to compare machine results to human ones. Therefore we created a game where the player is given with test images and has to classify them. The game was played by the authors and some friends (n<10) (you are encouraged to play too :-)). Of course, this amount is not representative but can give some sense. The results are shown in table 2. The results for the CNN are close but not similar to our reference. For Mobile Net we achieved 85% accuracy on the test set whereas the reference achieved 91%. The gap might be explained by some modifications they done that we didn't - adding a re-scaling layer at the beginning of the network and a global average pooling layer before the last FC. Self-supervised ViT shows the best results, followed by supervised ViT, Resnet, Mobile Net, and human. We continue the analysis with the best results - Resnet for CNN and DINOv2 for ViT. Confusion matrix results are shown in figure 3. ViT misclassified the classes in a similar manner whereas CNN misclassified Asian by approximately 3 times over the African. Visual results are shown in figure 4. The rightmost image in figure 4 shows a discrepancy between the provided label (Asian) and real one (African) meaning that not all test set were classified correctly. For creating CNN activation maps we used a built-in PyTorch library Gradient-weighted Class Activation Mapping (Grad-CAM) which highlights the regions that are important for making predictions [4]. We can infer that for high confidence images CNN highlights "important" regions, e.g., the ears and face. In opposite, for low confidence images it highlights many widespread regions that don't focus only on the objective. Examples of CNN activation maps can be seen in figure 5.

Model	Mobile Net	Resnet	supervised ViT	self-supervised ViT	Human
Test accuracy (%)	85	87	90	94	79

Table 2: Test Accuracy Results

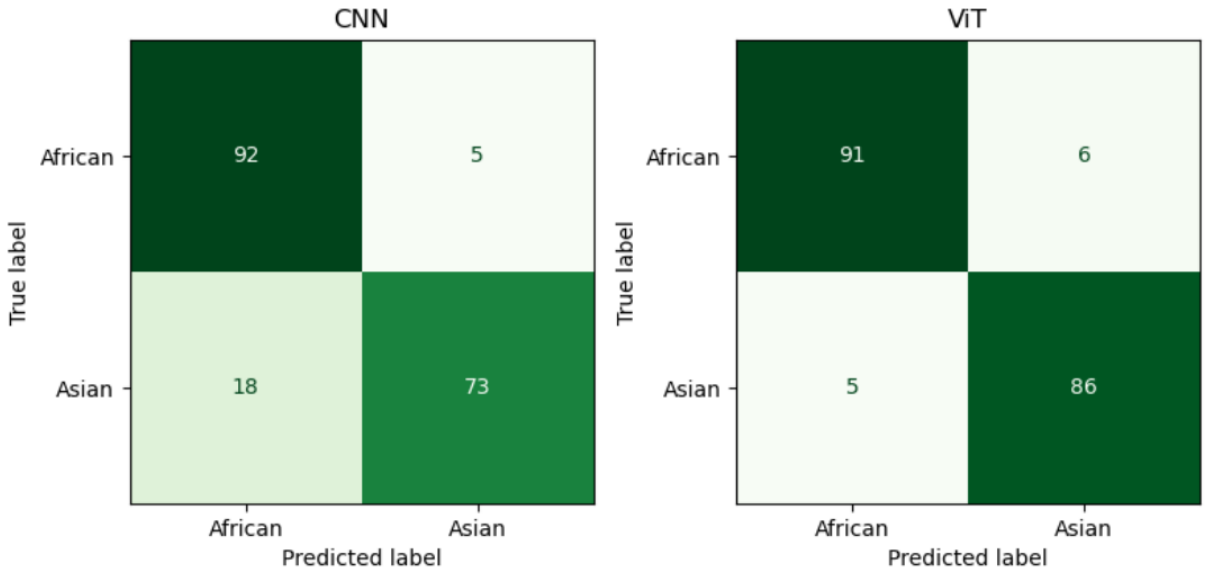

Figure 3: Confusion Matrix

6 Conclusion

Our results showed an improvement for ViT over CNN. ViTs offer better scalability and generalization capabilities when pre-trained on large amounts of data and then transferred to small image dataset benchmarks. As figure 6 suggested ViTs are already capable of paying attention to regions that are far apart right from the starting layers

Figure 4: Vision Results

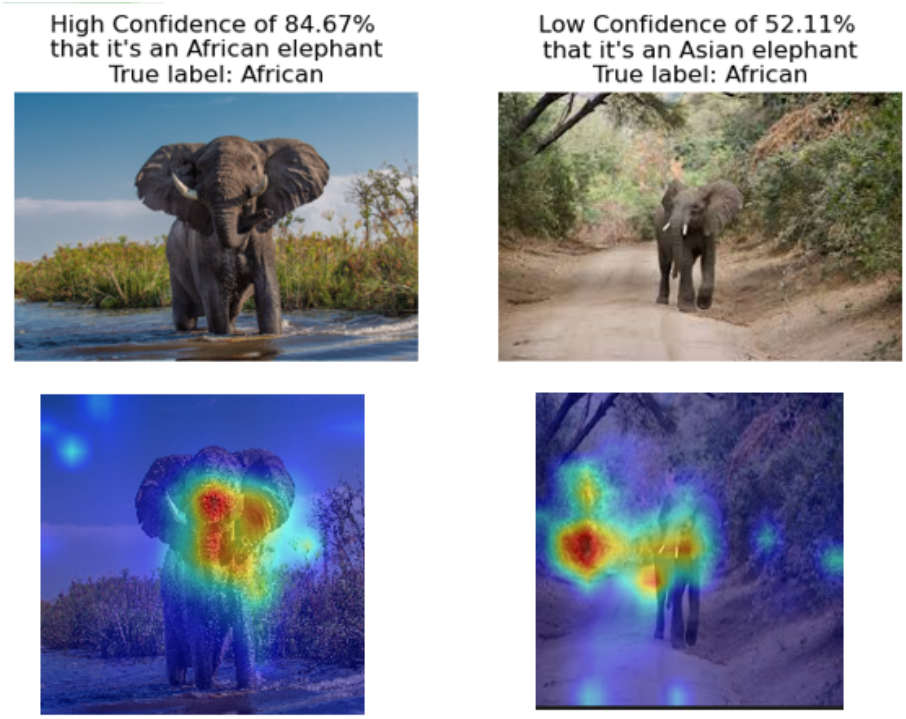

Figure 5: CNN Activation Maps

of the network, in opposite to CNNs which has a finite receptive field at the start. The self-attention mechanism allows ViTs to capture global and local dependencies within the image, enabling it to understand the contextual relationships between different regions [3]. However, Mobile Net is a lightweight architecture that has many fewer parameters compared to ViT (About 16 times) and it achieved competitive accuracy, showcasing its efficiency in resource-constrained environments. If we trained from scratch we would expect to get poor results due to a small dataset, but CNN's might outperform ViTs. This is due to the presence in CNNs of inductive biases, which helps them to grasp more rapidly the particularities of the images but makes it difficult to understand global relations. ViTs are free from these biases, which enables them to also capture long-range dependencies in images at the cost of more training data [7]. By examining the ViT models we infer that self-supervised model outperforms supervised one. This result was also seen in other works and we think it is because of better feature representation.

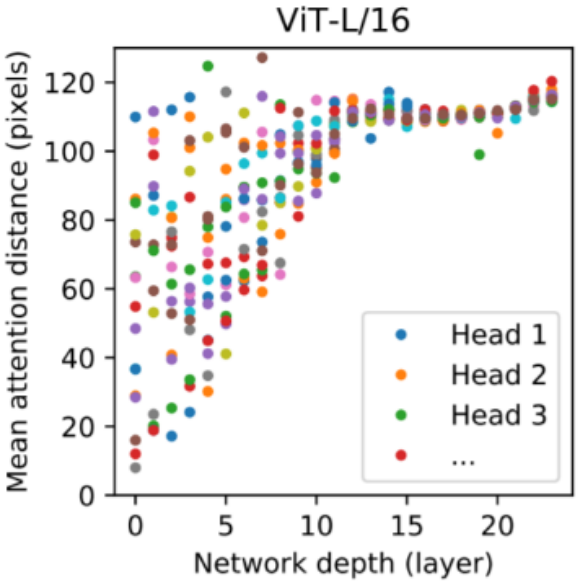

Figure 6: Attention distance with respect to the layers for ViT

7 Future Work

We suggest the following future work:

- Distinguishing different types of animals (e.g., tigers vs. lions, leopards vs. cheetahs).
- Exploring different pre-trained models for CNNs and ViTs.
- Investigating the effectiveness of ViTs on larger datasets with more diverse classes.
- Analyzing the robustness of ViTs against adversarial attacks.
- Integrating multi-modal learning to utilize both image and text data for better understanding and classification of images.
- Collaborating with wildlife conservation organizations to deploy the developed models for real-world applications, such as monitoring and protecting endangered species.

References

[1] ABHINAND. *Vision Transformer (ViT): Tutorial + Baseline*. https://www.kaggle.com/code/abhinand05/vision-transformer-vit-tutorial-baseline.

[2] Nasruddin Az. *Elephclass: Asian vs African Elephants Classifier*. https://www.kaggle.com/code/nasruddinaz/elephclass-asian-vs-african-elephants-classifier.

[3] EduardoSimon. *Fine-Tuning a Vision Transformer*. https://www.blend360.com/thought-leadership/fine-tuning-a-vision-transformer.

[4] Jacob Gil. *pytorch-grad-cam repository*. https://github.com/jacobgil/pytorch-grad-cam/tree/master.

[5] Luthei. *VIT vs CNN on WikiArt*. https://www.kaggle.com/code/luthei/vit-vs-cnn-on-wikiart.

[6] MetaResearch. *dinov2*. https://github.com/facebookresearch/dinov2.

[7] Vivmankar. *Asian vs African Elephant Image Classification Dataset*. https://www.kaggle.com/datasets/vivmankar/asian-vs-african-elephant-image-classification.