Machine Learning for Identifying and Classifying Egyptian Monuments

Course number: Stat 453, Spring 2025 | **Team number**: 14

Zachary Amsterdam, Sophomore undergraduate with a Double Major in Data Science and Biology | **Contact**: zamsterdam@wisc.edu

Samuel Pekofsky, Senior undergraduate with a Major in Data Science and Certificate in Computer Science | **Contact**: pekofsky@wisc.edu, sampekofsky@gmail.com

Carina Campbell, Sophomore undergraduate with a major in Data Science and Certificates in Classics and Leadership | **Contact**: carinacampbell2305@gmail.com

Brian Slupecki, Senior undergraduate with a Double Major in Data Science and Statistics | **Contact**: bkslupecki@wisc.edu

Abstract

This project explores the use of deep learning models to classify images of ancient Egyptian monuments, focusing on structures photographed from various angles and perspectives. We compare the performance of Convolutional Neural Networks (CNNs) such as ResNet-50 with Vision Transformers and hybrid models, aiming to evaluate their robustness to viewpoint variation and architectural similarity. Using the "egypt-monuments-dataset" available in Kaggle, we aim to address challenges such as sample size, limited viewpoint diversities, and class imbalance. Our approaches are broken down into two models, incorporating fine-tuning, data augmentation, and metric learning to improve classification accuracy. Through transfer learning and model comparison, we aim to identify which models generalize best under constrained conditions. This work contributes to cultural heritage preservation by testing machine learning systems on region-specific, low-resource datasets and highlighting the importance of model adaptability in real-world classification problems.

1. Motivation

For our project, we will focus on using deep learning to classify images of Egyptian Monuments from various angles. Many state of the art models are trained and tested on a large and varying dataset. The Egyptian Monuments we will focus on share many similarities and are all from the same region. Some are of figures, and share common body and face features, as well as similar clothing. Other monuments are pyramids, again very similar structures, while there are also temples which share many common building features. We knew our main challenge was going to be differentiating between these similar structures, especially considering that the pictures were from all different angles, and we had to deal with occlusion. However, while these monuments are similar, they are also all unique in their own way, and we believed we could design deep learning models to differentiate between them.

We hoped that our project would not only demonstrate Deep Learning's ability to classify Egyptian Monuments, but more broadly, Deep Learning's potential to distinguish similar structures from a common region if the right models and fine tuning are used. For our models, we used 70/15/15 train, validation and test split. As this is a classification model, we measured our models using accuracy, or the percentage of the time the model predicted the correct monument.

2. Existing Literature

The literature reviews we conducted examine three key approaches to improving deep learning based image classification in the context of visual similarity and data augmentation. Wei et al. (2022) proposes a GAN-based method for generating multi-view angles and data expansion. Their project focuses on grocery product recognition, aiming to expand their dataset to create a more comprehensive and robust model that remains unchallenged by angle changes. This approach offers a solution for augmenting underrepresented classes and improving the model–highly relevant for datasets like the Egyptian Monuments Dataset, where not every monument has equal representation across training epochs and iterations.

Arandjelović et al. (2016) introduces NetVLAD, a CNN-based embedding method, which excels in place recognition. This is particularly effective, as it focuses on image similarities as opposed to fixed class labels, which offers an alternative approach to combating subtle structural differences between monuments. This is especially relevant for Egyptian monument classifications as this offers a unique methodology to refine the model and improve potential weaknesses in monument similarities.

Furthering these ideas, Hesham et al. (2021) provides a survey of CNN advancements and underscores the architectures like ResNet50, which introduces residual connections to stabilize and deepen learning. This paper uses a small dataset of Indian Monument images (1,286 images total). At the time, they felt they did not have sufficient datasets for classifying Egyptian monuments, which was their overall project goal. They aimed to compare ML methods like KNN to Deep Learning models like ResNet-50 and VGG16 for their dataset. After multiple experiments and hyperparameter tunings, they found that ResNet-50, a DL method, had the best results, with almost 94% accuracy. This was especially relevant to the Egyptian Monument Dataset, as it demonstrates a stable approach to creating an image-recognition model.

3. Limitations of Existing Models or Methods

While the explored methods offer practical approaches for our model creation, they come with undeniable limitations. More broadly, image classification models have notable restrictions. Deep learning models, especially large ones like Vision Transformers, are often data-hungry, needing thousands of well-labeled images per class to perform well. This makes training from scratch difficult, increasing the risk of overfitting when attempting to fine-tune large-scale models.

Another notable challenge is viewpoint invariance. Many existing models are not built to naturally handle rotated, occluded, or angled views, unless they have been specifically trained on those variations. This becomes especially challenging for the Egyptian Monument Dataset, where monuments often share key architectural features like columns, engravings, and stone textures. This makes identification tricky for models that depend heavily on global visual cues. This may lead to misclassifications, especially when considering the varying perspectives of the monuments in the dataset.

Finally, there is an undeniable gap in the literature when addressing region-specific datasets. Landmark recognition work is historically focused on globally iconic sites like the Eiffel Tower of the Taj Mahal. There has been less focus on more similar, heritage-rich datasets like the Egyptian monument one, which often showcase more nuanced complexities in defining specific feature vectors.

3.1 Limitations of the GAN Model

Generative Adversarial Networks, otherwise known as GANs, are deep learning models that generate synthetic data while simultaneously training an opposing generator and discriminator. While GAN images may help combat small or limited datasets, they often contain vague visual artifacts that lack the finite details necessary for classifying complex structures like ancient monuments. This gap between real and synthetic data may limit classification accuracies, which may become problematic in this context.

3.2 Limitations of NetVLAD

NetVLAD is a CNN-based feature aggregation method which generates image embeddings by summarizing local descriptors into compact vectors. While this may be especially effective in place recognition, it does not perform direct classification, needing integration with other models or high levels of post-processing. This reliance on embedding similarity can also struggle when identifying highly similar structures that lack distinctive global features. This particular approach was incompatible with our dataset, which already contained specific class labelings.

3.3 Limitations of ResNet-50

ResNet-50 is a CNN with 50 layers that advantages residual connections to allow deep, stable learning. While it is efficient, effective, and often high-performing, it may experience execution plateaus when used on data with subtle class differences. This is especially problematic for the Egyptian Monument dataset, as one key goal is to combat difficulties the model may have in structural similarities. Additionally, ResNet-50 may not be able to capture long-range spatial dependencies as well as transformer-based models.

4. Dataset

The dataset being used in our project is the “egypt-monuments-dataset,” which was taken from kaggle. It is 222.69 MB in total and consists of 4782 total pictures, split between 22 ancient Egyptian monuments, giving roughly 217 images per monument. The minimum and maximum numbers of pictures for singular monuments are 68 and 410, respectively, which was offset using data augmentation. Each image in the dataset displays its monument from a different viewpoint or in a different lighting. The dataset is prelabeled based on the pre existing subdirectory of each image, but we had to reconfigure the data structure, as it was easier for the model.

4.1 Data Loading, Preprocessing, and Augmentation

The images in our dataset being used are all JPG, JPEG, and PNG files, so any other file type is filtered out. The images were then reorganized based on their monument name subdirectories to clean the dataset. Each image was resized to 256x256 and then cropped back down to 224x224, so they were all the same size. Then, they were randomly horizontally flipped, rotated 15º, and given a color jitter.

To make sure each monument was equally represented in the model, we balanced the dataset by oversampling the underrepresented classes. This created an equal number of images for each monument in the dataset.

4.3 Data Splitting

After the data had been preprocessed, it was split into the proper training, validation, and test subsets. This is necessary to prevent overfitting and to limit bias in the model. We split our dataset into the following distribution: Training = 70%, Validation = 15%, and Test = 15%.

DataLoader was utilized for creating batches and shuffling the training data for each iteration of the model being created.

5. Proposed Method

We will experiment with Convolutional Neural Networks (CNNs) and CNN-Transformer Hybrids, both of which are supervised learning models. These models were fine tuned through manipulation of various hyperparameters and architectures. In doing so, we hoped to learn how to manipulate our model's parameters and architecture to create a highly effective image classifier.

5.1 Convolutional Neural Networks

CNN models are often the default for image recognition, based on our exploration of existing work and literature. CNNs share weights and biases across hidden neurons in a layer, so they are more efficient and effective at learning the patterns in the inputs. There are two types of hidden layers in CNNs, being pooling and fully connected layers. Pooling layers reduce the dimension of feature maps, in turn reducing computational costs. Fully connected layers enable the model classifications to be made and then processed to the output layer. CNNs implement backpropagation, which adjusts the weights based on the difference between the predicted and actual outputs.

5.1.1 Our CNN Model

Our CNN model uses ResNet-50 as its backbone. We chose to use ResNet-50 due to its exceptional ability in feature extraction, and ability to inhibit the vanishing and exploding gradient problem. Additionally, ResNet-50 is pretrained on the ImageNet dataset, which is a very large and well-established dataset for image classification. ResNet-50 has 50 layers in total, the first 49 of which are frozen. This is due to the pretraining on ImageNet, so the early layers are already tuned to many general features like edges, textures, and basic shapes.

We updated the classifier in the general ResNet-50 model, so it would be better tailored towards our dataset. This new classifier consists of the following: Linear → ReLU → Dropout → Linear. We use a ReLU activation function because that is the typical activation function implemented with the ResNet-50 model. Additionally, a Dropout layer is added to prevent overfitting, with the dropout rate being set to 0.3 after preliminary testing.

CrossEntropyLoss was then used as the loss function, since it is commonly used in image classification models. This is because it can quantify the difference between the model's predictions and true class labels. After further research, we decided to use Adam as an optimizer, since it can improve model efficiency and accuracy. Adam incorporates adaptive learning rates and momentum, so the optimization will converge faster than other optimizers.

In total, our CNN model is 51 layers, with the 49 frozen layers in the ResNet-50 backbone, and 2 additional layers in our classifier. There are 512 hidden neurons in the modified fully connected block, and 22 neurons in the output layer.

5.2 Hybrid CNN-Transformers

Hybrid CNN-transformer models are able to combine the advantages of both models. The advantages of the CNN model are outlined above in section 5.1. Transformers are able to implement attention, so each token is able to directly influence the other tokens in the input. Additionally, they allow for parallel processing, enabling the model to learn and calculate multiple weights simultaneously, speeding up the training process. Transformer models succeed best in capturing global trends, while CNNs are better at extracting local features in the input.

5.2.1 Our Hybrid Model

The backbone of our hybrid CNN-transformer model is EfficientNet-B0. This is a pre-trained CNN from the ImageNet dataset specialized for image classification problems, making it appropriate for our project. It is the smallest of its model family, best suited for our small compute ability compared to more sophisticated models.

EfficientNet-B0 backbone has a total of 231 layers. We customized the model once again to be better suited towards our dataset. We added two multi-head self attention blocks to the model. These blocks allow the model to focus on different aspects of the input sequence simultaneously, providing a faster and more complex understanding of the data. This is then put through an average pooling layer and finally a fully connected layer, in order for the class predictions to be made. We did not update the classifier in this model, but could have benefited from that implementation, if time permitted.

As was done in the CNN model, we used CrossEntropyLoss as the loss function and used Adam as an optimizer. This allows for the only difference between the models to be their components, rather than how they are measured, allowing an appropriate comparison between the two.

Our hybrid CNN-transformer model has a total of 237 layers: 231 coming in the EfficientNet-v0 backbone, four in the multi-head self attention blocks, with one pooling, and one fully connected layer. There are a total of 1,280 hidden neurons in this model.

5.3 Hyperparameters

We ran many different variations of each model by changing the hyperparameters. This was done by changing the following hyperparameters with their respective possible values used: learning rate = 0.001, 0.005, batch size = 16, 32, 64, and number of epochs = 50, 200. All possible combinations with these hyperparameters were put into models.

5.4 Measurements

We were primarily concerned with testing accuracy, since it roughly measures the strength of the model. Additionally, we compared training and testing accuracy to assess overfitting. Moreover, this would keep model runtime to a minimum without significantly impacting testing accuracy. Therefore, validation loss also proved to be important–but mainly for the sake of early stopping and choosing the best model. Furthermore, since we experimented with different model structures, batch sizes, learning rates, and training epochs, we compared both accuracy and runtime.

6. Compute Budget

Our models were trained and assessed using a computer with an Nvidia RTX 3070. The training of the 12 variations of the ResNet 50 model took about 11 hours, while the training of the hybrid model took 7 hours for the 12 variations. The difference in the time each took could be because the hybrid model had less epochs as well as early stopping. Each of the 12 variations for each model type had different parameters. Compute power also was not a limiting factor for us, as it appears more epochs did not improve results. When working on the ResNet50 model, it did not improve when using 200 epochs rather than 50. For the hybrid model, when the max epochs was set to 100, it always stopped early.

7 Results

When hyperparameters were tuned effectively, both of our models performed quite well, achieving well over 90% accuracy in many instances. With this being said, we trained 12 variations of both models, with several possible combinations of epochs, batch size, and learning rates, and there was a wide variation of accuracy for many of the variations, especially those with a higher learning rate.

7.1 ResNet50 Model Results

For our experimentation with our ResNet50 model, we trained and tested our model with all 12 combinations of either 50 or 200 training epochs, batch sizes of 16, 32, and 64, and learning rates of 0.001 and 0.005. Please view figure 1 below.

learning rate	training epochs	batch size	test accuracy
0.001	50	16	93.2692%
0.001	50	32	93.9560%
0.001	50	64	94.6429%
0.001	200	16	95.4670%
0.001	200	32	95.8791%
0.001	200	64	96.0165%
0.005	50	16	84.3407%
0.005	50	32	85.3022%
0.005	50	64	90.3846%
0.005	200	16	87.3626%
0.005	200	32	90.1099%
0.005	200	64	89.2857%

Figure 1, ResNet50 Experiment Results

learning rate	training epochs	best epoch	batch size	test accuracy
0.001	30	29	16	97.91%
0.001	100	8	16	97.07%
0.001	30	13	32	97.77%
0.001	100	4	32	96.09%
0.001	30	29	64	98.61%
0.001	100	13	64	97.91%
0.005	30	11	16	3.07%
0.005	100	10	16	31.38%
0.005	30	30	32	41.14%
0.005	100	24	32	37.24%
0.005	30	3	64	8.37%
0.005	100	65	64	92.61%

Figure 2, Hybrid Experiment Results

As can be seen above, there is roughly a difference of 8% in test accuracy between our best and worst performing versions of our ResNet50 model. Having a learning rate of 0.001 yields better results across the board, however, the number of training epochs and batch size seems to be much less influential, though, there is a very small trend of improvement as both of these are increased.

7.2 Hybrid Model Results

Similar to our experimentation with our ResNet50 model, we also tested 12 hyperparameter combinations with our hybrid model, however, we instead used 30 and 100 epochs with early stopping, since this model proved more computationally demanding and took longer per epoch. Figure 2 will include the results below, however, it also includes the epoch at which the best model was found, since this model was run with early stopping, and therefore the epoch at which its best version was found is valuable information.

Similar to the ResNet50 model, the hybrid model performs better with a learning rate of 0.001 as opposed to 0.005, however, the differences in performance are far more significant. Our intuition was right to use fewer epochs with early stopping, as the best epoch column shows that more training would have been ineffective, however, this model also shows far more sensitivity to changes in batch size, especially with a learning rate of 0.005. Additionally, Due to the very poor results in 5 of the 6 variations with a learning rate of 0.005, we suspect a vanishing or exploding gradient, once again demonstrating one area where the ResNet50 model comes out on top.

7.3 Class-Performance Comparison

To look at class performance and where our models may have been getting mixed up, we looked at confusion matrices, with one example from the hybrid model below (figure 3). We see the clear diagonal, showing that there was not any class that our model significantly struggled to predict. Also when looking at its rare incorrect predictions, it connects back to what we discussed in our motivation, the similarities between monuments. Incorrect predictions for pyramids were almost all predicting other pyramids. The same can be said for statues and temples. For example, the class that was the least accurate was the Menkaure Pyramid. It incorrectly predicted the Bent Pyramid twice, and the Khafre Pyramid 4 times. While this is one example of our confusion matrices, all other matrices for both our ResNet models and hybrid models were similar and reflected these patterns.

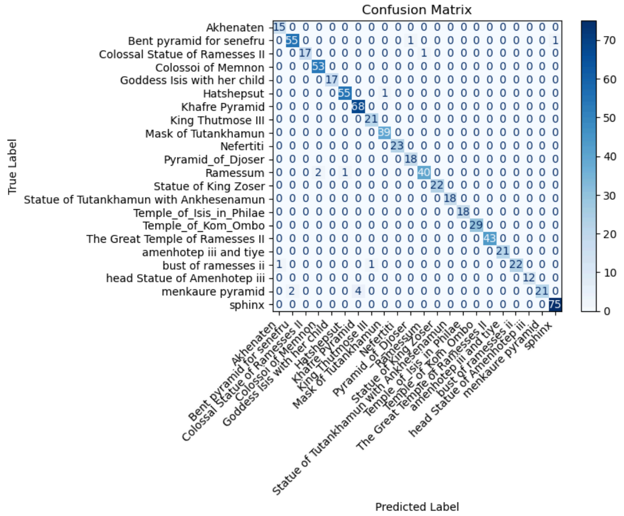

Figure 3, Example Hybrid Model Confusion Matrix

8. Limitations and Future Work

Our project did achieve our goal of creating a functional deep learning model capable of classifying ancient Egyptian monuments, even with image augmentation and occlusion. It also demonstrates the broader ability of deep learning to distinguish between visually similar structures–when sufficient fine-tuning and model design are implemented. Despite this accomplishment, our model did have undeniable limitations.

For one, the model developed was relatively lightweight, containing few trainable parameters. This likely limited its ability to learn highly nuanced features. Future research should explore more advanced architectures and conduct broader hyperparameter optimization to enhance accuracy and generalizability. Promising models include YOLOv5, EfficientNetV2, and multilayer perceptrons, (MLPs). In addition, future experiments could benefit from a further exploration of learning rates, epoch counts, and batch sizes to further refine model performance.

8.1 Limitations of our Hybrid Model

A major limitation of our hybrid model is its complexity. While its architecture has the potential to yield favorable accuracy results, it also makes it far more computationally intensive when compared to standard CNNs. The hybrid model also experienced extreme levels of performance variation and sensitivity to hyperparameter changes. As previously mentioned, we suspect a potential vanishing or exploding gradient issue to be the root cause of this problem. This means without intensive hyperparameter training, the model is simply not reliable. Altogether, while our hybrid model has the potential to perform well, it requires sufficient fine tuning and potential architecture changes to reach optimal performance.

8.2 Dataset Limitations

Our dataset is fairly small by image recognition standards, consisting of just under 5,000 images across 22 classes. It is also quite imbalanced between classes, with certain monuments being represented by as few as 68 images, while others over 400. Our oversampling of smaller classes increases the chances of overfitting on these classes, potentially artificially boosting accuracy on minority classes. These imbalances and limited viewpoint diversity present challenges, which become especially prevalent when evaluating performance on unseen data. These factors may skew the results of the data or generalize its performance. Future work should consider additional data augmentation techniques or the collection of more balanced datasets to support broader applicability.

9. Conclusion

We were happy with our models overall ability to accurately predict Egyptian Monument Images. Through our use of ResNet50 and hybrid models, we were able to consistently achieve accuracy in the 90% range. This shows that Deep Learning models can be used to classify region specific images that share many similarities, as well as handle different angles and occlusion. We hope this spurs on further research into this sector of image classification.

References

1. Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., & Sivic, J. (2016). **NetVLAD: CNN architecture for weakly supervised place recognition**. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5297–5307. https://doi.org/10.1109/CVPR.2016.575
2. Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. (2017). **Large-scale image retrieval with attentive deep local features**. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 3456–3465. https://doi.org/10.1109/ICCV.2017.372
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. (2021). **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale**. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2010.11929
4. Liu, Z., Lin, Y., Cao, Y., et al. (2021). **Swin Transformer: Hierarchical Vision Transformer using Shifted Windows**. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 10012–10022. https://doi.org/10.1109/ICCV48922.2021.00989
5. Weyand, T., Araujo, A., Cao, B., & Sim, J. (2020). **Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval**. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2575–2584. https://doi.org/10.1109/CVPR42600.2020.00265
6. Detone, D., Malisiewicz, T., & Rabinovich, A. (2018). **SuperPoint: Self-Supervised Interest Point Detection and Description**. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. https://arxiv.org/abs/1712.07629