Recognizing Artwork Attributes with

The Metropolitan Museum of Arts

Anuj Sarda, Nitin Limhan, Preethi Bharathy, Srija Dasgupta, Vedant Ghate

December 06, 2022

Abstract

The Metropolitan Museum of Art in New York usually referred to as The Met, is a collection of approximately 1.5 million pieces, of which over 200 thousand have been digitized and illustrated. The iMet Collection is a dataset for fine-grained artwork attribute recognition curated using images of museum objects in The Met's collection. The annotations in iMet are verified by SMEs around the globe. The labels generated in iMet characterize the object from the viewpoint of art history and are often oblique in describing minute characteristics from a visitor's perspective. In this project, we analyzed the iMet and narrowed down the requirement to a classification problem. Our project work focuses on providing fine-grained attributes for museum objects.

1. Introduction

Most of the early research so far is based on instance retrieval and mid-level attributes like color, and person detection. In recent years, computer vision research on the lines of fine-grained visual classification (FGVC) has become increasingly popular. The rapid progress of deep neural networks has enabled computer vision algorithms to capture powerful representations for such complex semantic problems. Leveraging this capability, it is now possible to implement models performing coarse-grained category recognition. Using this advancement, [1] presented a dataset, named iMet Collection 2019, which contains fine-grained and research-grade attributes for the images in the Met. Subject Matter Experts (SME) around the world helped in generating online cataloging information for the digital images in the Met. The iMet Collection is developed using this information (the research grade attributes). It contains a wide range of data attributes including multiple object classifications, artist, title, period, date, medium, culture, size, provenance, geographic location, and other related museum objects. While the SME-generated annotations describe the object from an art history perspective, they can also be indirect in describing finer-grained attributes from the museum visitor's understanding.

2. Method

To identify the fine-grained attributes for each image and solve this classification problem, different models were explored and trained on the iMet dataset.

- **Initial Architecture**: A binary classifier built using CNN that classifies if the given image belongs to a culture/tag attribute. The output of the primary model is used to decide where to send the images to - **Secondary model 1** (identifies all the attributes that describe the image under the culture label) or **secondary model 2** (identifies all the attributes that describe the image under the tag label).
- **Pre-trained Model 1**: Multi-label classification using Multi-Layer Perceptron (MLP) with pre-trained model DenseNet121.
- **Pre-trained Model 2**: Multi-label classification using Multi-Layer Perceptron (MLP) with pre-trained model EfficientNet

2.1 Initial Model

The initial model's architecture can be found in Figure 1. The primary model has the below layers (three convolution layers and four fully connected layers):

- Input Layer: 3 * 128 * 128
- Output Layer: 1 * 2

Each of these layers also has a Maxpool of 2, padding of 1, and stride of 1 added to the convolutional layers. The activation function used func1 is used in each layer. The output layer returns 0 or 1 for each image based on which attributes are more dominant (culture/tag). If culture attributes are dominant for a given input image, then the image is classified as 0 or else 1.

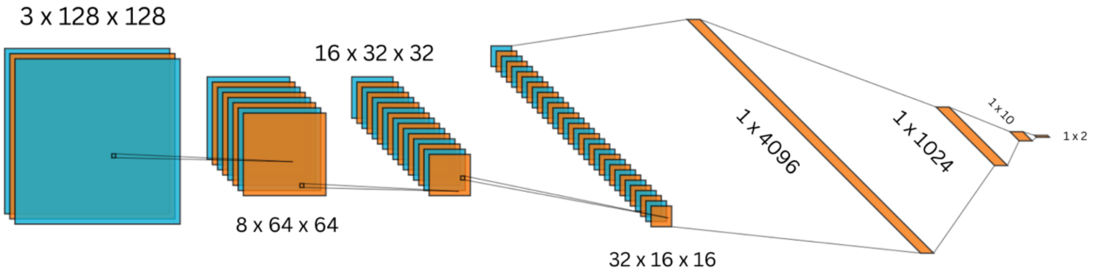


Figure 1 CNN based Binary Classifier Architecture

The secondary models for identifying all the cultural attributes and the tag attributes based on the primary model have the following architecture (refer to Figure 2).

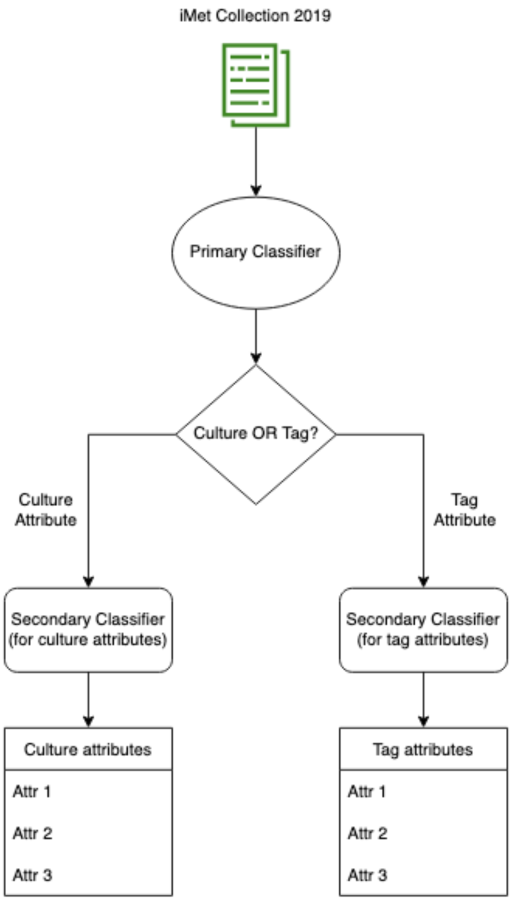


Figure 2 Multiclass classifier

2.2 Pre-trained Models

Implementing the initial model architecture approach resulted in certain data challenges (elaborated in the Discussion Section). Therefore, upon further literature survey, the next best step was to perform multi-label classification using pre-trained models, a popular approach among the competitors in the iMet 2019 Kaggle challenge.

The pre-trained Models 1 and 2 were adopted and re-implemented using this Kaggle reference - https://www.kaggle.com/code/hidehisaarai1213/imet-pytorch-starter/notebook?scriptVersionId=12405149

2.2.1 Model 1

Model 1 uses pre-trained CNN DenseNet121 to perform multi-label classification along with Multi-Layer Perceptron (MLP). This in-built pre-trained model has 120 convolutions and 4 AvgPool. The MLP layers defined on top of the DenseNet model have the below structure:

- Input Layer: 1024 * 1024
- Output Layer: 1024 * 1103
- Number of folds: 5 (40 Epochs)

2.2.2 Model 2

Model 2 uses pre-trained EfficientNet CNN to perform multi-label classification along with Multi-Layer Perceptron (MLP). The MLP layers defined on top of the EfficientNet model have the below structure:

- Input Layer: 1024 * 1024
- Output Layer: 1024 * 1103
- Number of folds: 5

DenseNet121: Also known as Densely Connected Neural Networks is used for increasing the depth of CNs. The main motivation behind using DenseNet is to overcome the problem caused when building deeper CNNs. As the information path between the input layer to the output layer increases, the information can vanish before reaching the output layer and this issue is addressed when using DenseNet.

EfficientNet: A CNN with a scaling method that uniformly scales all the dimensions – length/breadth/height using a compound coefficient. EfficientNet models typically outperform previous CNNs in terms of accuracy and efficiency, resulting in smaller parameter sizes.

2.3 Other failed model attempts

We also explored training CNNs using Keras which uses Adam optimizer. A re-implementation was adopted from Kaggle.

3. Experiments

3.1 Dataset

The dataset primarily consists of 100K+ training images and 1103 unique image attributes. A sample of the images that are found in the training set is shown in Figure 3. As seen, one of the dimensions of the image is 300. The training data contains the image ids and the attributes associated with them. Each image has multiple attributes associated with it and each attribute describes the image. For example, **id: 1 attribute: 51 616 734 813**. In the above example, for image id 1, there are 4 attributes associated with it {51, 616, 734, 813} which describes the image.

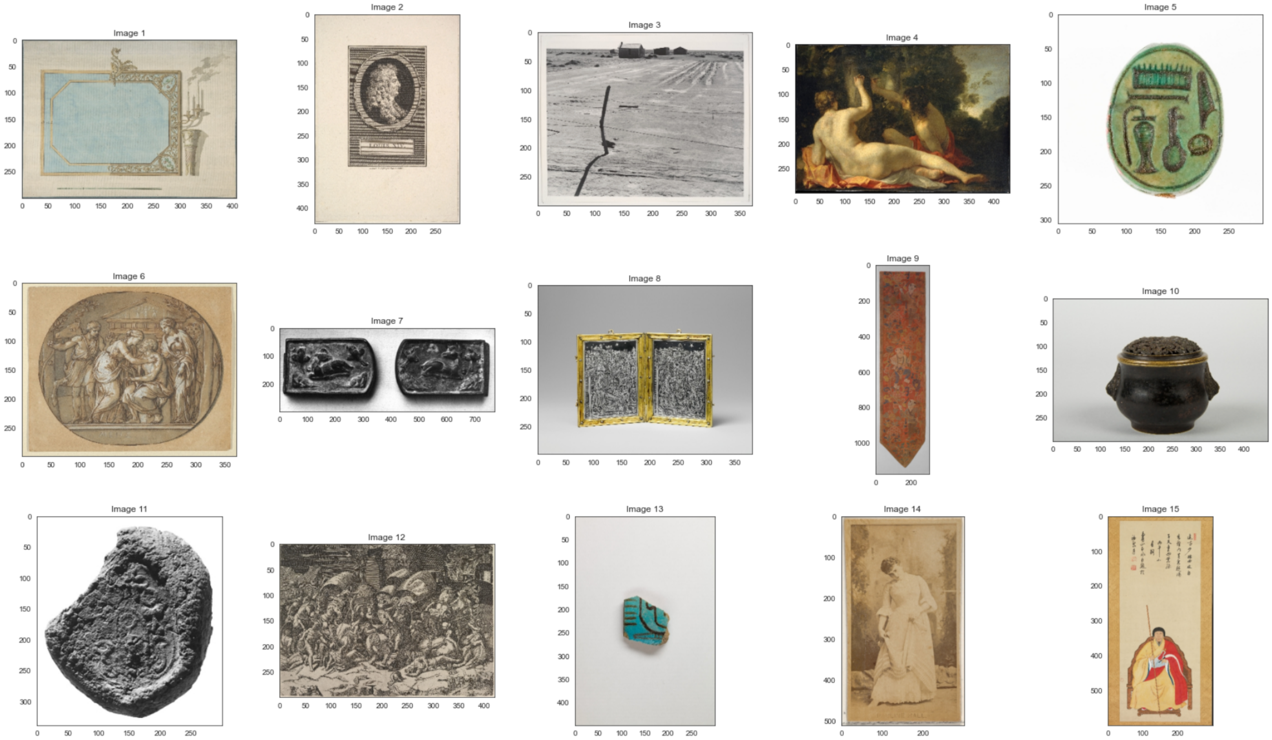


Figure 3 – View of sample images

The 1103 unique attributes are divided into culture and tag (398 cultures, 705 tags). For the same example, **id: 1 attribute: 51 616 734 813**, the attribute values below or equal to 398 are culture attributes, and the values above 398 and less than 1103 are tag attributes. The dataset being tag dominant (tag attributes > culture attributes), results in images having more tag attributes than culture attributes. For a given image, the minimum number of attributes is 1 and the maximum number of attributes is 11. The average number of attributes that describe an image lie between 2 and 3 (Figure 4).

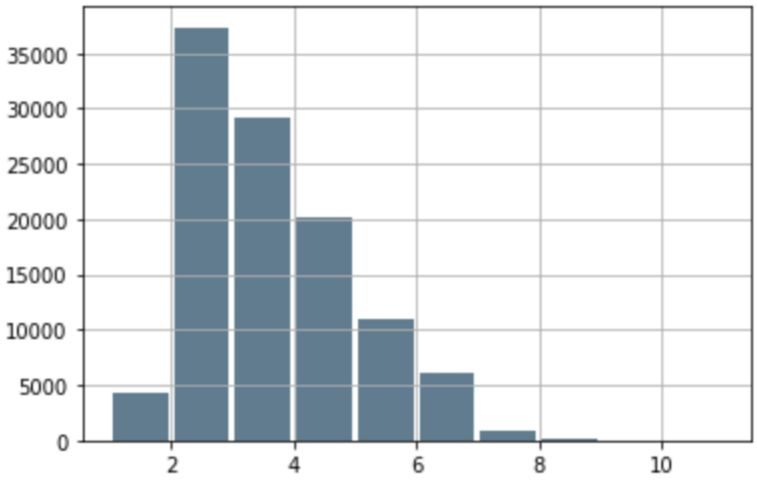
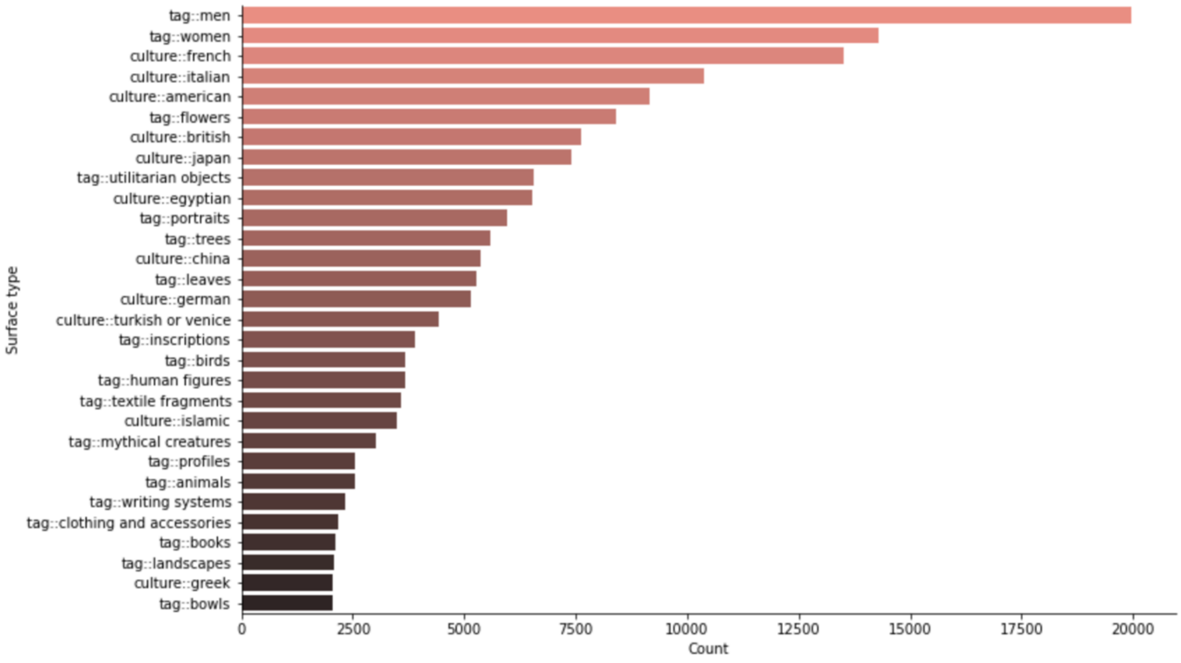


Figure 4 – count the number of attributes that describe an image

Upon diving deep, the top 30 attributes popularly used to describe images and the bottom 30 or the least used attributes to describe images can be found in **Figure 5**.



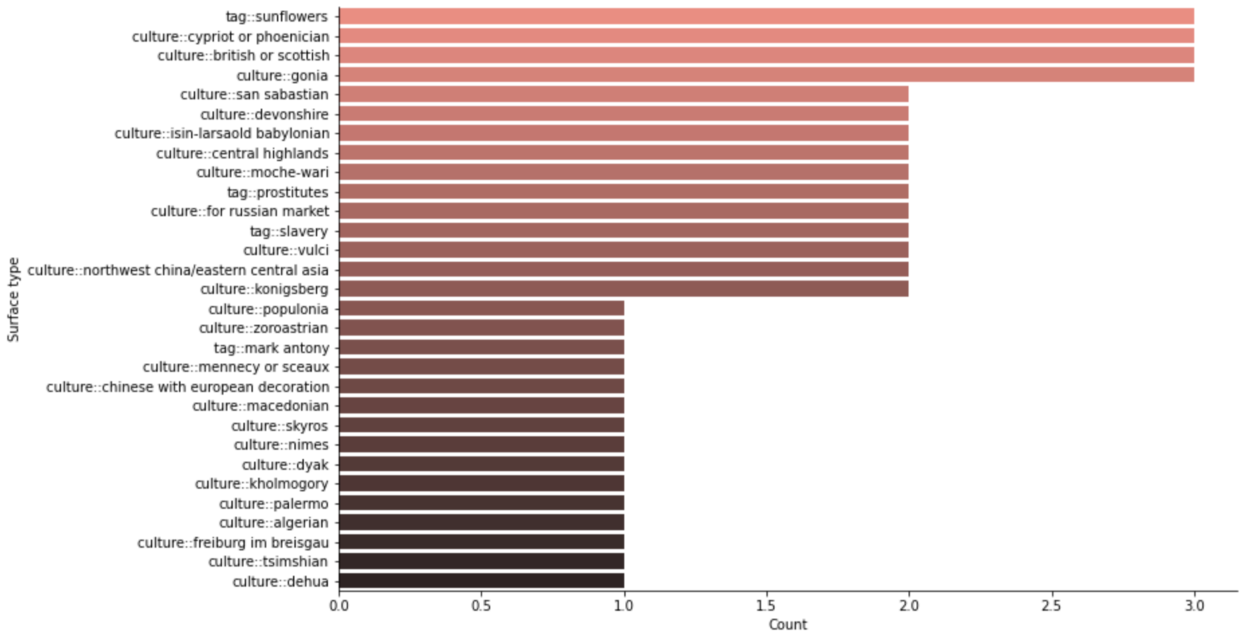


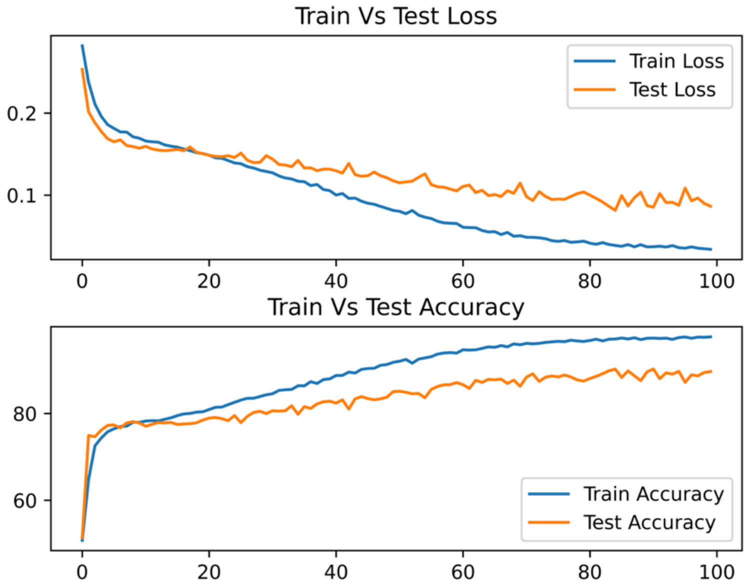
Figure 5 – Top 30 and bottom 30 frequently found attributes used to describe images

From the above visualizations (Bottom 30 and Top 30 mainly) for the 1103 unique attributes that describe a given image, the dataset is highly imbalanced. For instance, the bottom 30 labels that are assigned to the image are found only 1 to 3. These attributes add to the difficulty of training the model to learn and predict all these 1103 labels effectively. Additionally, the imbalance is also seen as the attributes are classified into two categories (Tags and Culture attributes). The tag attributes are found twice as much when compared to the culture attributes among images.

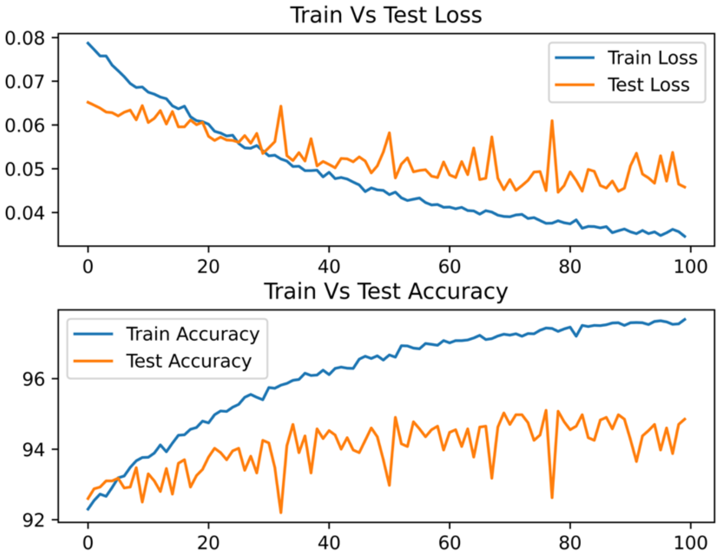
3.2 Results

We performed two major experiments, first with the CNN-based binary classifier and another one to study the combination of CNN with multilayer perceptron. Following are the results of our first experiment:

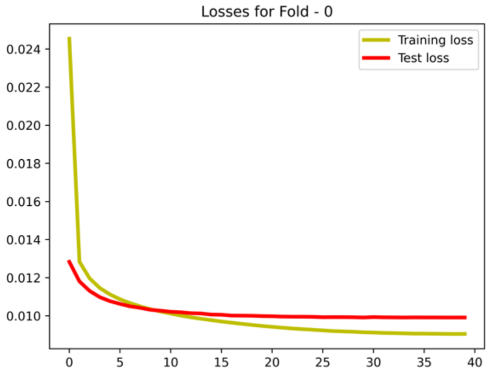
- Learning Rate – 0.0001, Batch size – 128 for 100 epochs
- The test accuracy was up to 90%

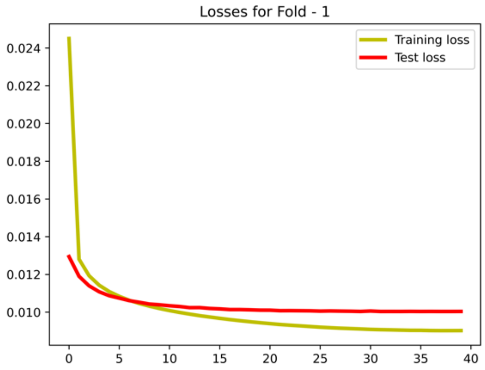


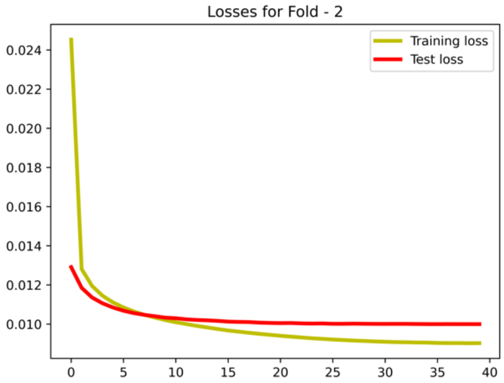
- Learning rate – 3e-05, Batch size – 128 for 100 epochs
- The test accuracy was up to 95%

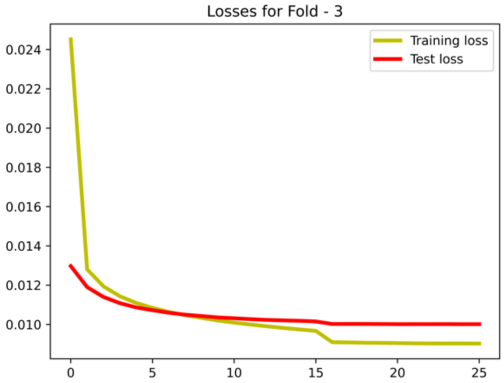


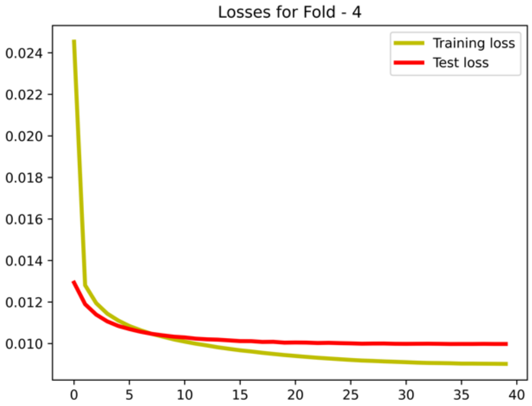
Following are the results we observed for the combination of DenseNet with multilayer perceptron. With this, we observed an F-score (evaluation metric) of 0.45.











4. Discussion and Analysis

Initially, the proposed architecture was as shown in figure 1. The primary model (binary classifier) was used to identify if an image is dominated by culture/tag attributes. The output of this primary model was used to decide which model to run next. The secondary model for culture/tag predicts the attributes under culture/tag respectively for a given image. The culture model predicts attributes that are identified in the image between 1-398 and the tag model predicts attributes that are identified in the image between 399 – 1103. The main reason behind this architecture was to address the imbalance found in the dataset due to the number of culture and tag attributes associated with an image.

Although the primary model was successfully able to learn the dominant attribute (culture/tag) for a given image, this approach wasn't ideal for the classification problem we were trying to solve. We ran into the below challenges:

- **Data Augmentation:** The initial analysis that we performed on the data indicated a mix of culture-dominated attributes and tag-dominated attributes. Depending on these, we designed the first model which was a CNN-based binary classifier. However, in a later analysis performed on the data, we noticed that each image would have at most 4 culture attributes and multiple tag attributes. This discovery invalidated our initially proposed architecture of using a binary classifier.
- **Data Imbalance:** Although this is still a classification problem where we are to identify multiple attributes for a given image, the data imbalance hampers the performance of the model.
- **Variable Ground Truth Tags**: The ground truth for each image is not uniform, an image can have 1 ground truth or just one attribute describing it and another image can have 4 or 5 ground truths (or attributes) describing it.

The above challenges led us to research more optimal approaches to solve this problem and directed us towards using Dense Neural Networks instead, to perform multi-label classification. Considering the limitations of computational capacity to train a Dense Neural Network from scratch, we moved in the direction of exploring pre-trained CNN models found in PyTorch. We studied and analyzed the combination of CNN with Multilayer Perceptron, with pre-trained models such as the DenseNet121 and EfficientNet. We also tried to re-implement these combinations but were restricted due to the available computational capabilities.

5. Conclusion

In this project work, we implemented a binary classifier with the use of convolutional neural networks and observed its accuracy to be pretty good over the iMet dataset. Following that, we studied various approaches for a multi-layer, multi-class classification required for datasets such as iMet. We explored the combination of CNN and Multilayer Perceptron and found it to be efficient. Based on our literature survey, we also believe that transformers can solve this problem in an efficient way.

6. Contribution

Anuj Sarda

- Hyperparameter tuning for primary model (to improve model accuracy)
- Hyperparameter tuning for pre-trained model 1 (to improve model accuracy)
- Pre-trained model 2 (EfficientNet) training
- Pre-trained model 2 (EfficientNet) with varying MLP layers
- Documentation and reporting

Nitin Limhan

- Model development for primary classification
- Experimenting with existing models and understanding the architecture
- Pre-trained model 1 (DenseNet) with varying MLP layers
- Visual representations and result gathering for report
- Documentation and reporting

Preethi Bharathy

- Exploratory data analysis
- Primary model training and monitoring
- Pre-trained model 1 (DenseNet) Monitoring and plotting results
- Hyperparameter tuning for pre-trained model 2
- Documentation and reporting

Srija Dasgupta

- Problem definition
- Data pre-processing scripts
- Documentation and reporting
- Report integration

Vedant Ghate

- Studying model architectures
- Literature Survey to find optimal approaches
- Pre-trained model 1 (DenseNet) training
- Pre-trained model 2 (EfficientNet) Monitoring
- Documentation and Reporting

7. References

[1] Zhang, C., Kaeser-Chen, C., Vesom, G., Choi, J., Kessler, M., and Belongie, S. The imet collection 2019 challenge dataset, 2019.

[2] Nguyen, V., & Kim, S. Cleaning and Structuring the Label Space of the iMet Collection 2020. ArXiv, abs/2106.00815, 2021

[3] https://www.kaggle.com/code/hidehisaarai1213/imet-pytorch-starter/notebook?scriptVersionId=12405149

[4] https://medium.com/the-artificial-impostor/notes-imet-collection-2019-fgvc6-part-1-91fa77a92435

[5] https://www.kaggle.com/competitions/imet-2019-fgvc6/overview