An Automated Approach to Artist Classification Using DL With the Help of Classical Principles. AKA Tim.

Alireza Alizadeh

Models and Principles

We use 3 different pretrained models and finetune them with our data while using a pretrained ResNet50 model with only its output layer changed as baseline (The change is necessary for the model to be fitted with our classes). The models in use are the following:

Architecture	Version	Parameters	Model Size (Approx.)	GFLOPs (Speed)	Primary Strength
ResNet	ResNet-50	**~25.6M**	~98 MB	4.1	The industry standard; very stable.
EfficientNet	B0	**~5.3M**	~20 MB	0.39	Incredible efficiency;
ConvNeXt	Tiny	**~28.6M**	~110 MB	4.5	Modern CNN with "Transformer-like" performance.

Each model then is trained on our preprocessed database and evaluated.

The Idea is to train models on even ground and compare one smaller and one bigger model to our Baseline. Training parameters and pipeline will be discussed in the training section.

Training[1]

Training was done on a preprocessed set of images from a total of 31 artists. Images all had been resized to 224×224 with the use of padding in our data preparation phase. The reason behind using padded images instead of simple resizing is to preserve the proportions in the painting. Note that this comes at a cost of models needing to learn that the black bars around the images are irrelevant data which can be a problem for smaller models.

The training was done on 20,80 splits for the test and train sets with the following parameters with an early stop mechanism which would terminate the loop if the model did not approve for 5 consecutive epochs:

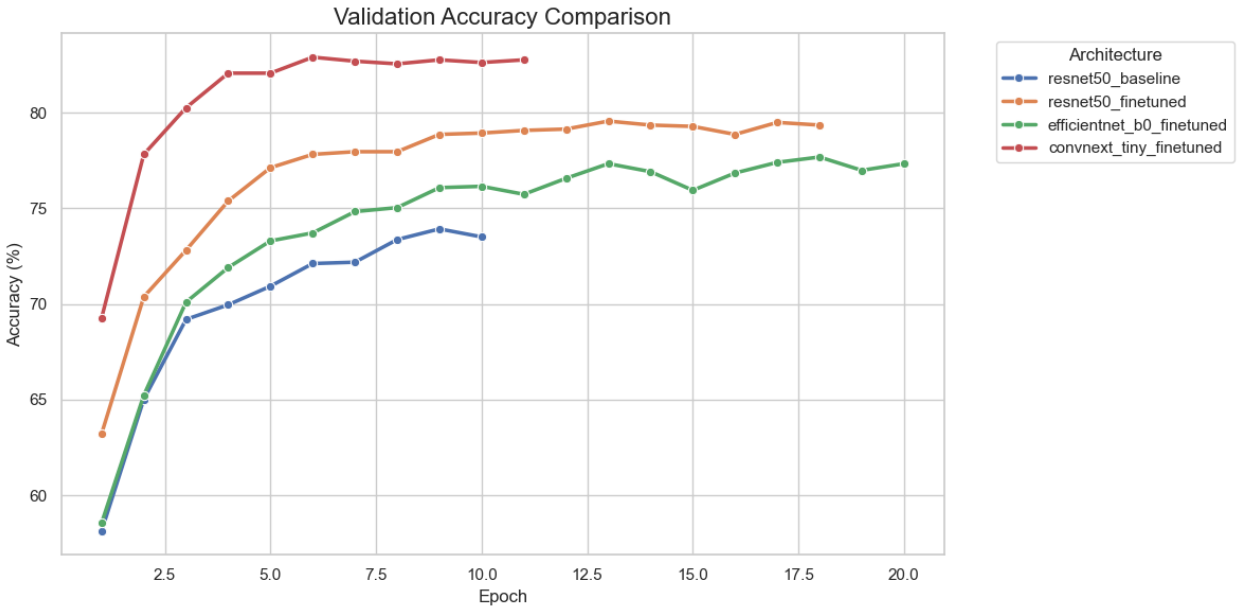
Parameter	Value
RandomHorizontalFlip	True
RandomRotation	10
ColorJitter	0.1, 0.1, 0.1
Normalize	[0.485, 0.456, 0.406], [0.229, 0.224, 0.225]
Optimizer	Adam (Early layers: 1e-6, Intermediate layers: 1e-5, Final layer: 1e-3)
Batch size	32
Epochs	20*

[1] All the results can be found in the training_results.ipynb located in the notebooks folder

*Baseline fitting was done on 10 epochs

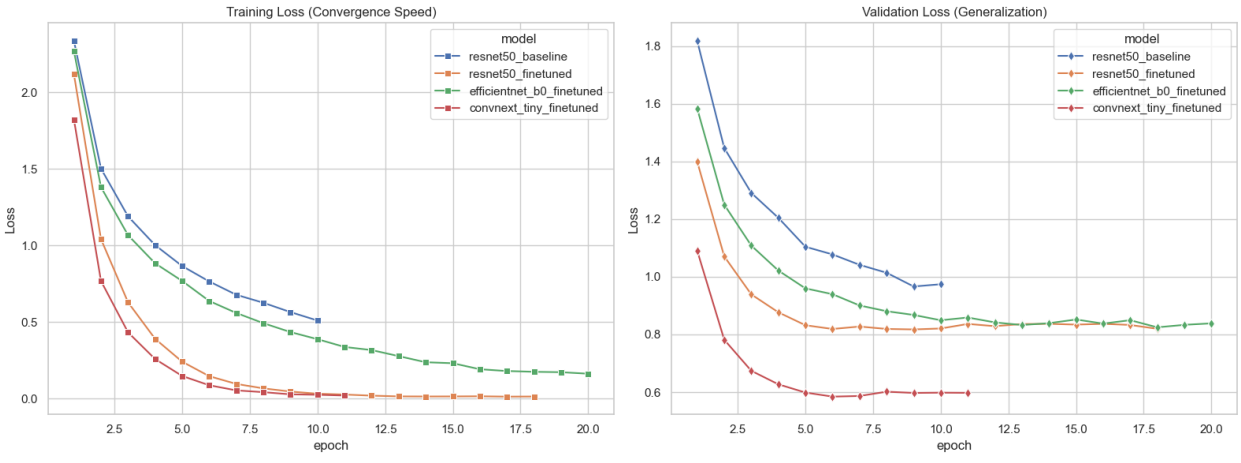
As expected, our bigger models performed better while finetuned but the interesting part is that EfficientNet-B0 while being significantly smaller does not fall behind as much as one would expect.

Accuracy analysis



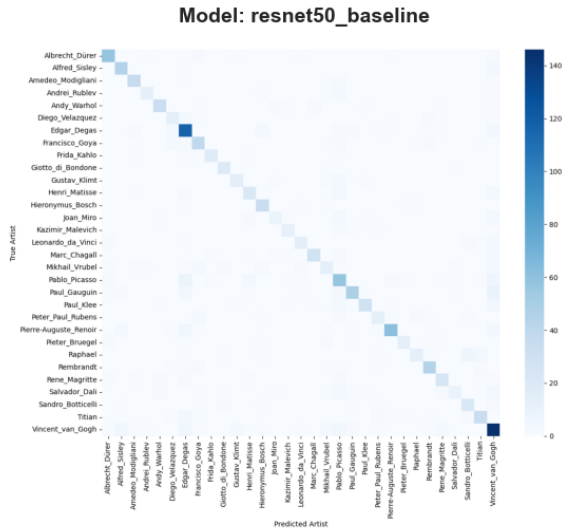
While our models hold an upward trajectory our limited resources stop us from training the models furthermore. Perhaps the most interesting take away from this plot is that our EfficientNet-B0 while being significantly smaller still outperforms the baseline and is extremely competitive with larger models.

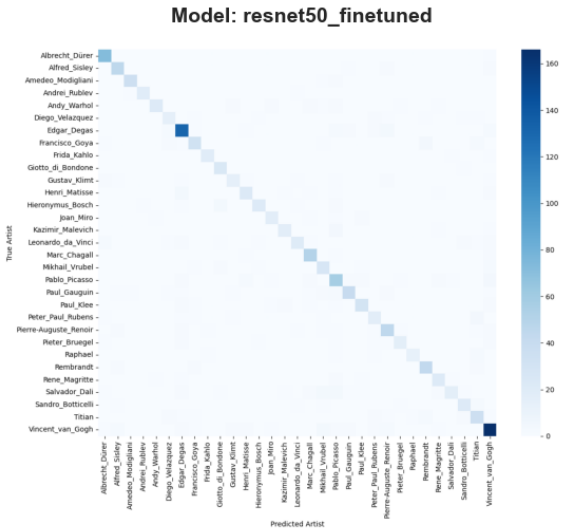
Loss Analysis

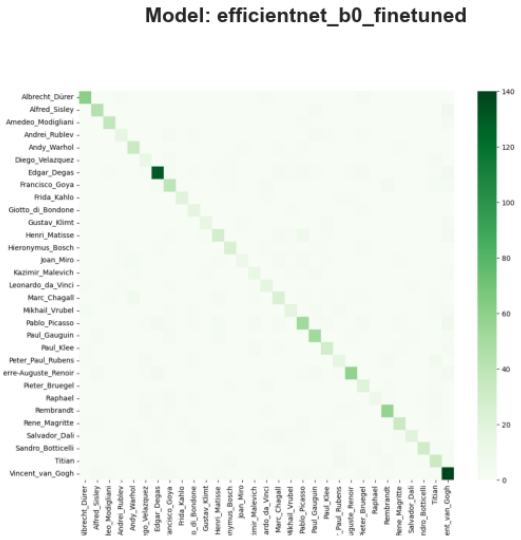


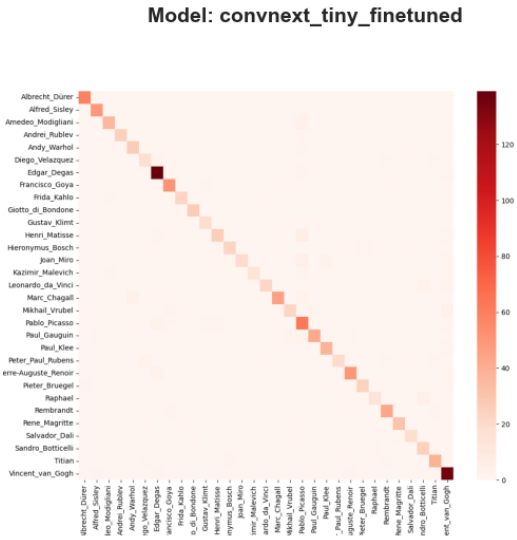
The same trend continues where we see improvements but can not keep training because of our resources. However, the metrics achieved in this limited time our still fairly acceptable. As expected, larger models have the edge but not by much.

Confusion Matrices and Misclassification Analysis









Other than baseline confusing some artists with Van Gogh due to large number of paintings by him other models seem to handle false positives fairly well. While some artists with similar styles and/or disproportionate number of paintings get mixed up sometimes resulting in false positives, models have respectable F1 scores as demonstrated below in the metrics section.

Metrics

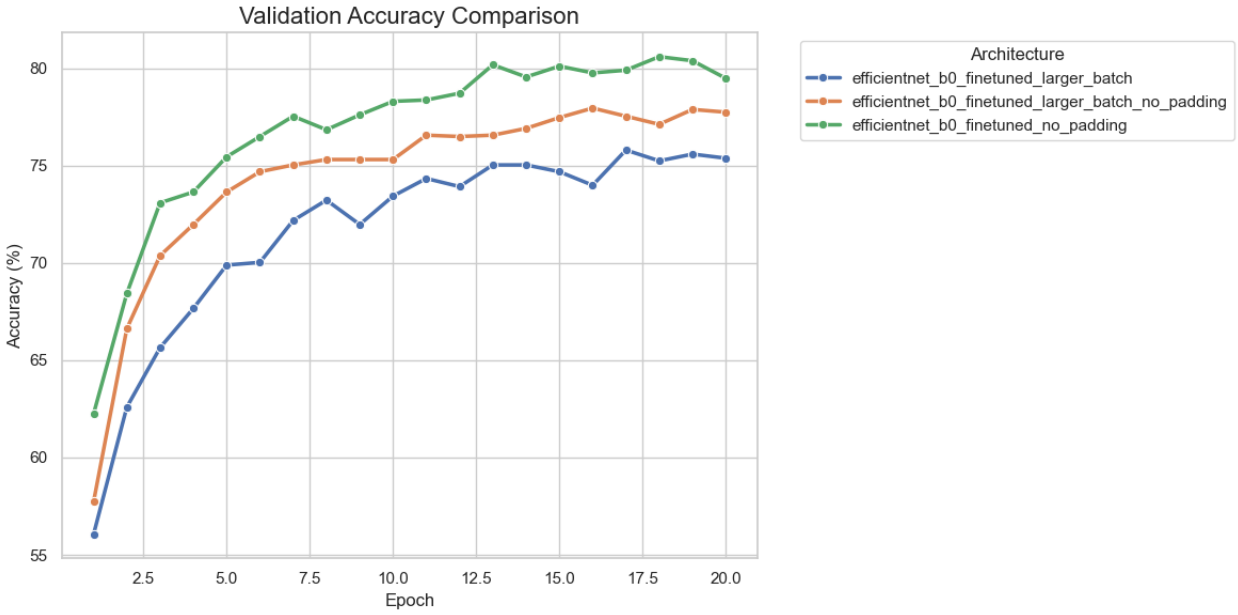
The following metrics were produced by the Bootstrap method to generate a sample of our database with 20% of its size.

Model	Accuracy	Precision	Recall	F1
resnet50_baseline	0.794159	0.811330	0.794159	0.792832
resnet50_finetuned	0.911683	0.914526	0.911683	0.911204
efficientnet_b0_finetuned	0.880389	0.888231	0.880389	0.879374
convnext_tiny_finetuned	0.943672	0.945681	0.943672	0.943143

Experiments[2]

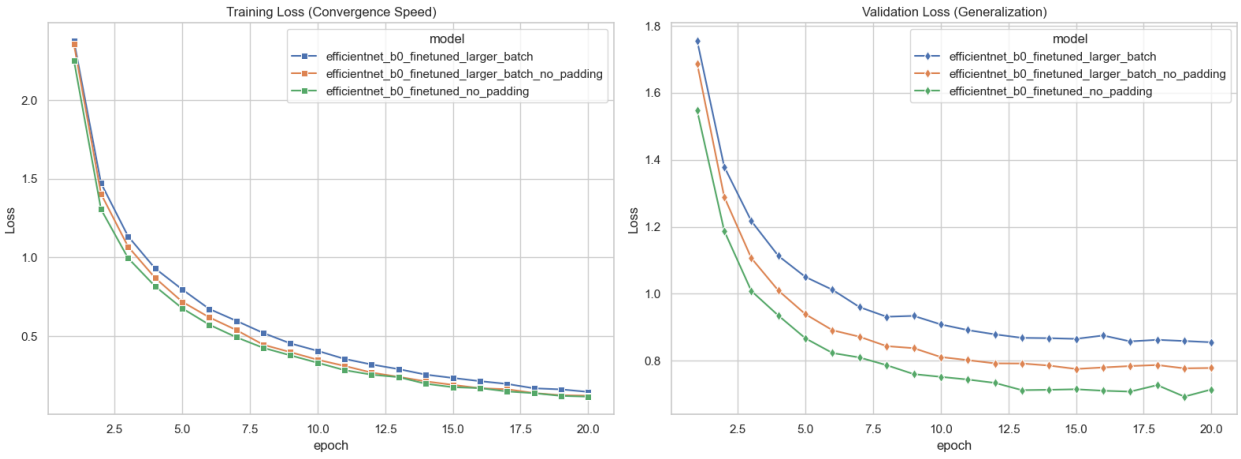
A number of experiments were done on our smallest model (EfficientNet-B0) in an attempt to improve its performance. These experiments were done with a combination larger batch size and a use of non-padded resized images. Even though the metrics did Improve their generalization remains questionable since parameters were catered to the models liking and might have resulted in overfitting.

Accuracy Analysis



As expected, small models prefer the no-padding approach since there is no useless data in the input.

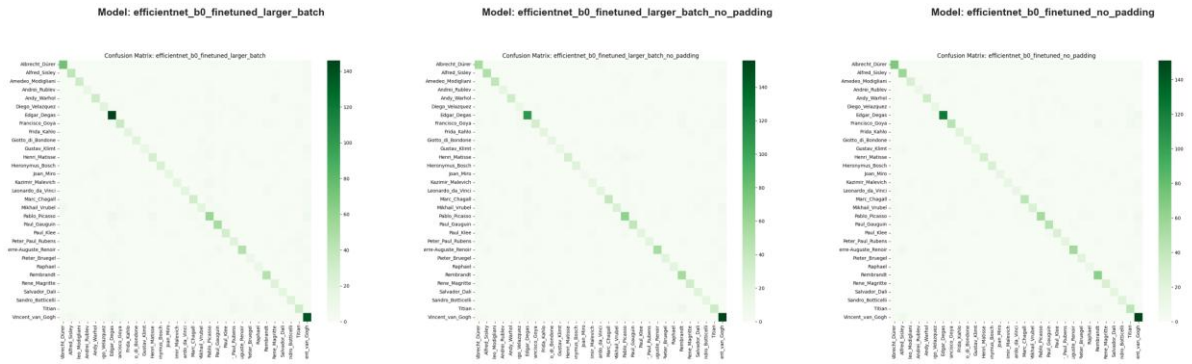
Loss Analysis



Interestingly training loss doesn't seem to be affected by our changes to the training process.

[2] All the results can be found in the experiments_results.ipynb located in the notebooks folder

Confusion Matrices and Misclassification Analysis



Models' misclassification rates don't change significantly compared to each other or our main model.

Metrics

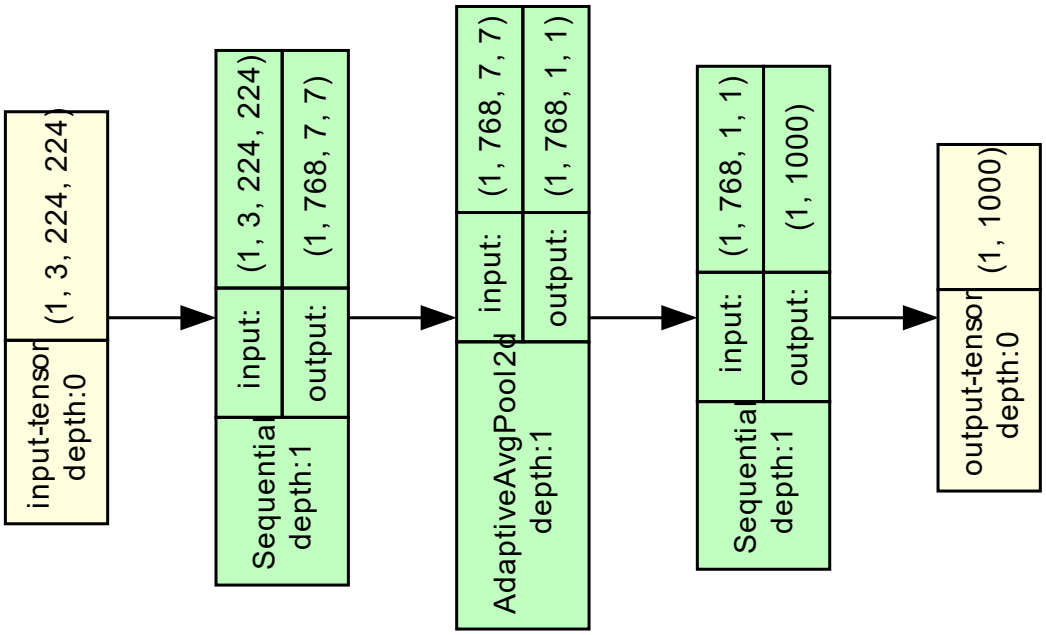
The following metrics were produced by the Bootstrap method to generate a sample of our database with 20% of its size.

Model	Accuracy	Precision	Recall	F1
efficientnet_b0_finetuned_larger_batch	0.872740	0.877499	0.872740	0.869706
efficientnet_b0_finetuned_larger_batch_no_padding	0.958971	0.960004	0.958971	0.958610
efficientnet_b0_finetuned_no_padding	0.965925	0.966484	0.965925	0.965896

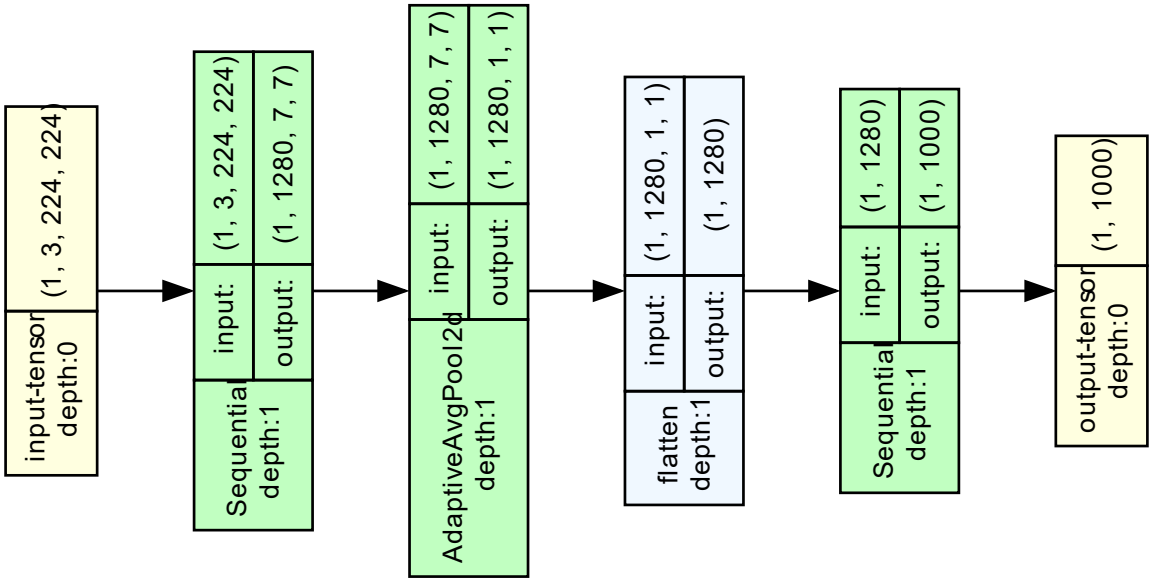
Model Structure

Following structures were generated using Torchview at the depth of 1 and for the following Input (1(Batch),3(RGB Channels),224(Image dimension X),224(Image dimension Y)).

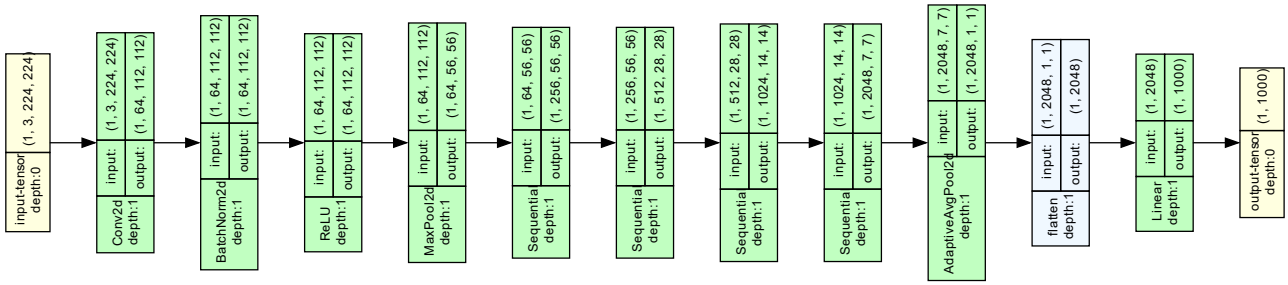
Convnext Tiny:



EfficientNet-B0:



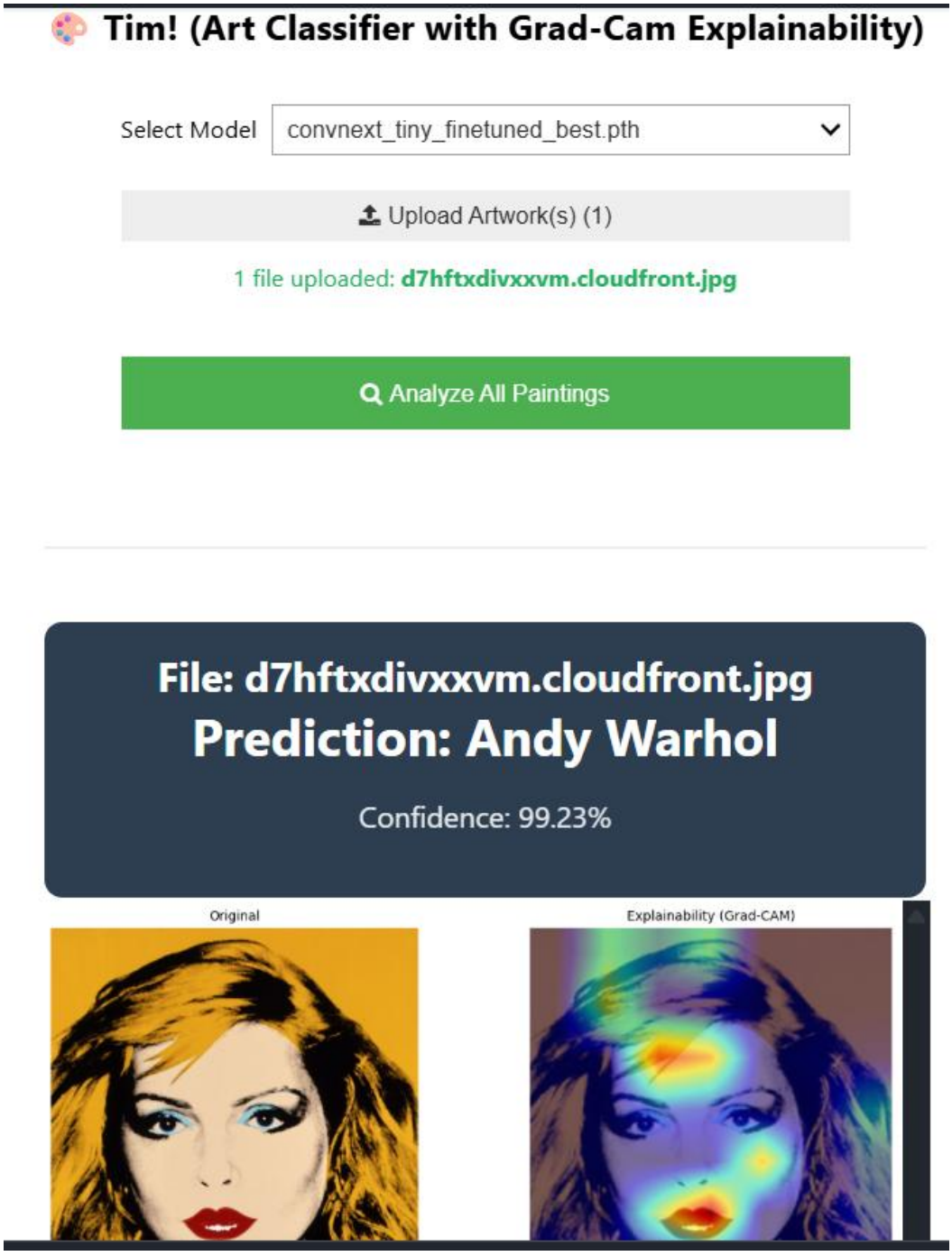
ResNet-50:



Demo

An interactive demo can be found in the notebooks folder in a file called “interactive_demo.ipynb”

Demo example:



Conclusion

While AI and Machine Learning performance are extremely dependent on computational resources and date, The ability to train and run models locally is somewhat neglected whilst increasingly important subject matter. Specially as privacy concerns raising day by day.

Having relatively small but performant models that run locally and can satisfy our day-to-day needs is a huge benefit for the end consumer both in terms of finances and privacy.

While our biggest models still outperformed the smaller ones EfficientNet-B0's performance being better than our baseline is a positive outcome and shows smaller models have potential and with proper finetuning and optimization can become very competitive with larger models.