Google Landmark Recognition Challenge

Adrian Iordache, Florin Gogianu

Faculty of Mathematics and Computer Science, University of Bucharest, in partnership with Bitdefender

Introduction

In the Spring of 2018, Google released a new dataset and a challenge on Kaggle, which requires a classifier being able to recognize and categorize landmarks. But the whole purpose of the challenge is not only to find a methodology that satisfies the requirements, the idea behind this dataset represents the need of Computer Vision society for understanding and developing more robust image classification systems.

Objectives

Finding a proper methodology for image classification on a subset of Google Landmark Dataset and establishing a baseline convolutional neural network architecture to be later used on bigger subsets of the initial dataset with focus on understanding and visualizing different perspectives of the problem.

Dataset Description

Taking into consideration the dataset size of approx. 1.2 million images and the number of 14500 classes we choose to train a baseline model on a subset of the most frequent 10 classes and after that we can go further with the number of classes.

As you can see below even a subset of 50 classes is highly imbalanced, so we may need to give thought to methods for balancing the dataset distribution, that like subsampling or some weighted loss function, but this it may not be necessary because even the lowest frequent class in the dataset has two thousand samples.

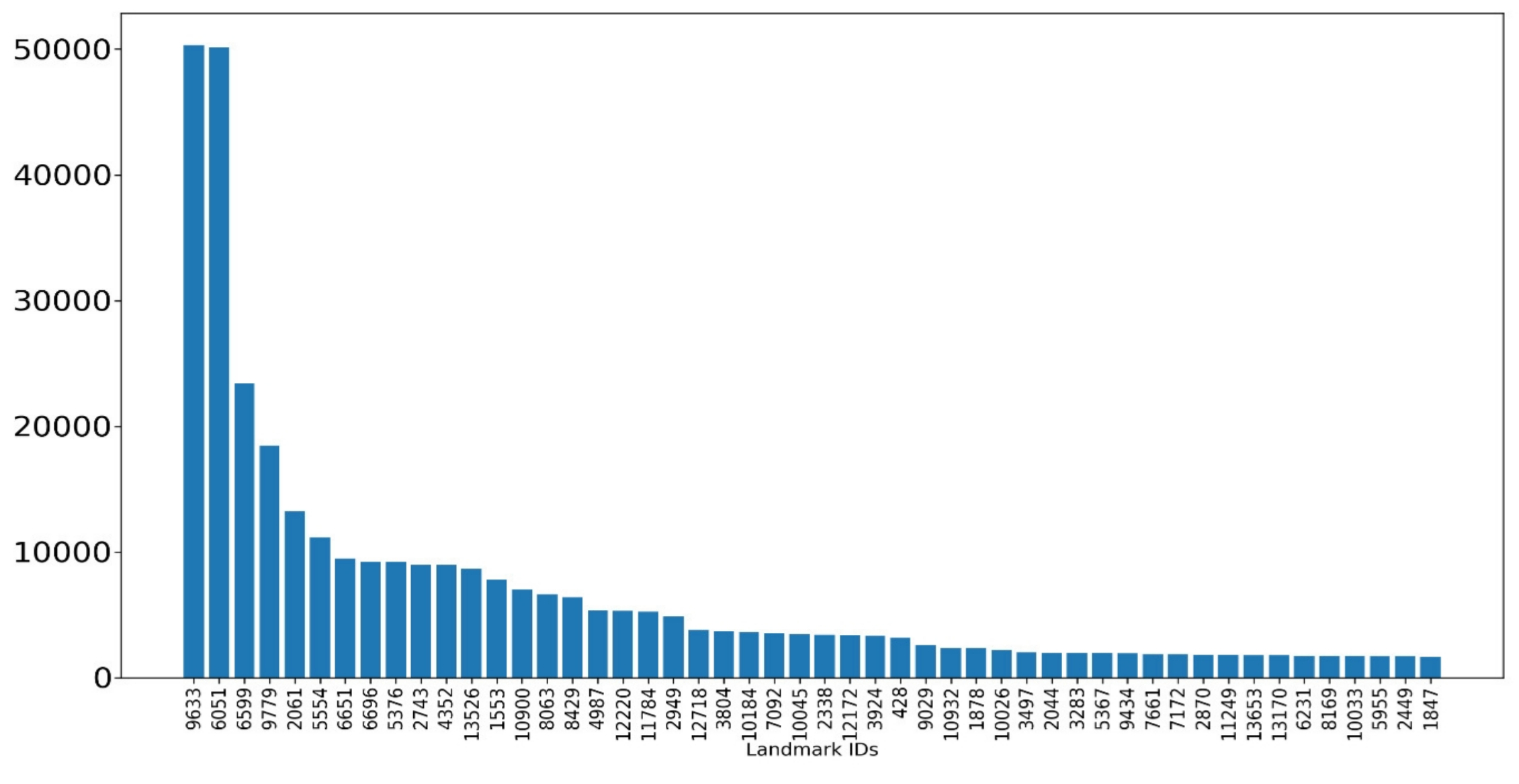

Fig. 1:Dataset Distribution with the most frequent 50 classes

Models Architecture

As we said, we will start on a 10 classes subset which contains approx. 202 thousand samples. As splitting methodology we will first subtract a 1% from the dataset as a holdout set and after that the rest of the samples will be splitted into train and validation sets, respectively 80% and 20%. For training, we choose to go head to head with two pretrained models on ImageNet, a ResNet-18 and a VGG-16.

Hyperparameters Optimization

Hyperparameters tried during our experiments:

- Learning rates: 0.01, 0.005, 0.001
- Epochs: Between 1 and 10
- Optimizers: Adam, SGD
- Batch Sizes: 16, 24, 32, 48, 96

Validation Scores

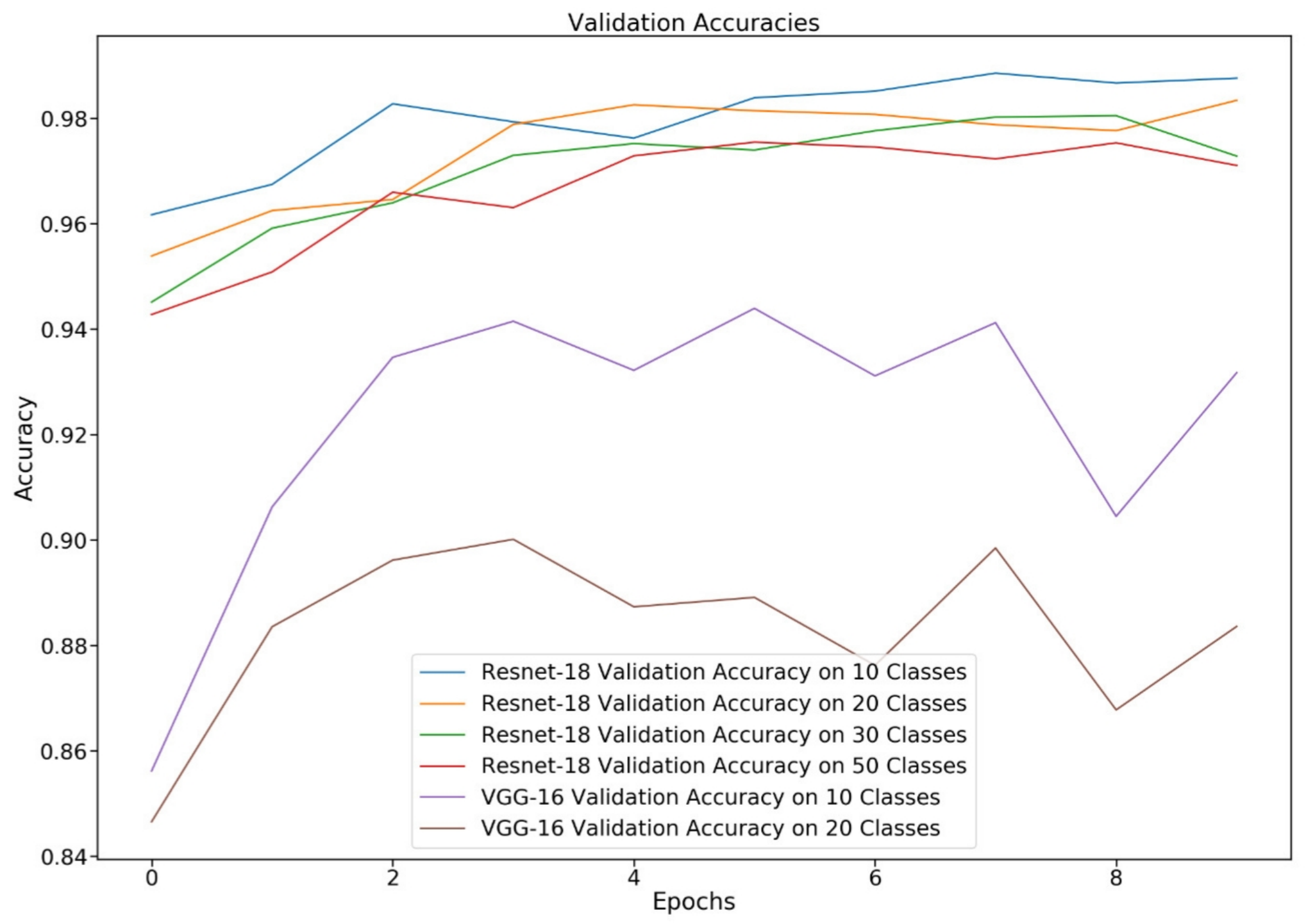

Fig. 2:Validation Accuracies

Augmentation Methodologies

- Random sized crop
- Random horizontal and vertical flip
- Random rotations of 90 degrees
- Keeping the features extractor weights frozen

It's worth mentioning that neither of our final models uses any type of augmentation or frozen weights.

Holdout Set Results

Model	No. classes	Train	Validation	Holdout
ResNet-18	10	0.992	0.988	0.988
ResNet-18	20	0.993	0.983	0.984
ResNet-18	30	0.991	0.980	0.969
ResNet-18	50	0.984	0.975	0.968
VGG-16	10	0.935	0.943	0.941
VGG-16	20	0.889	0.900	0.881

Interpretation of Results

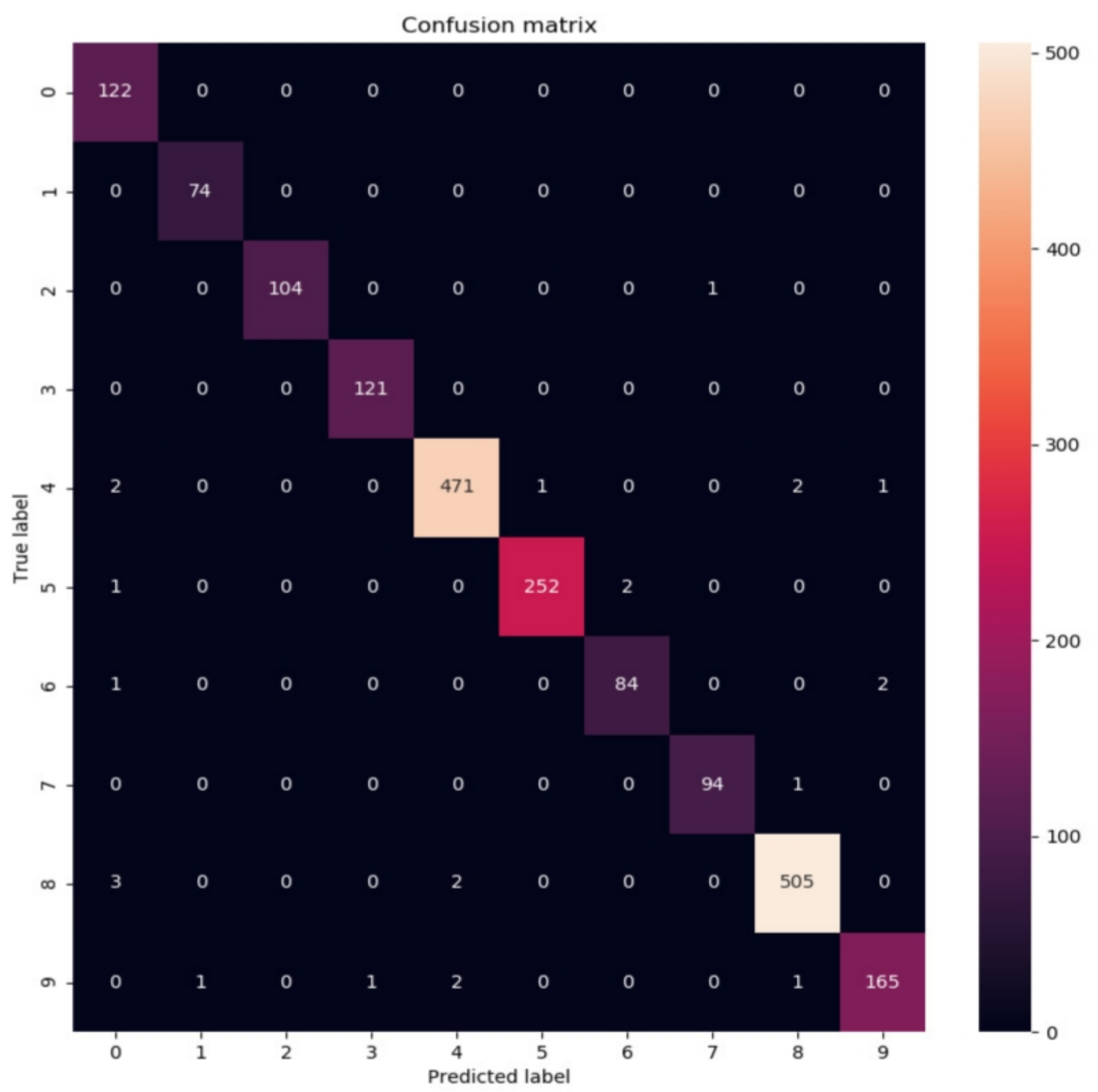

Fig. 3:Confusion Matrix for ResNet-18 with 10 classes

Our final models are trained over 10 epochs using as loss function CrossEntropyLoss minimized by an Adam Optimizer with a learning rate of 0.001.

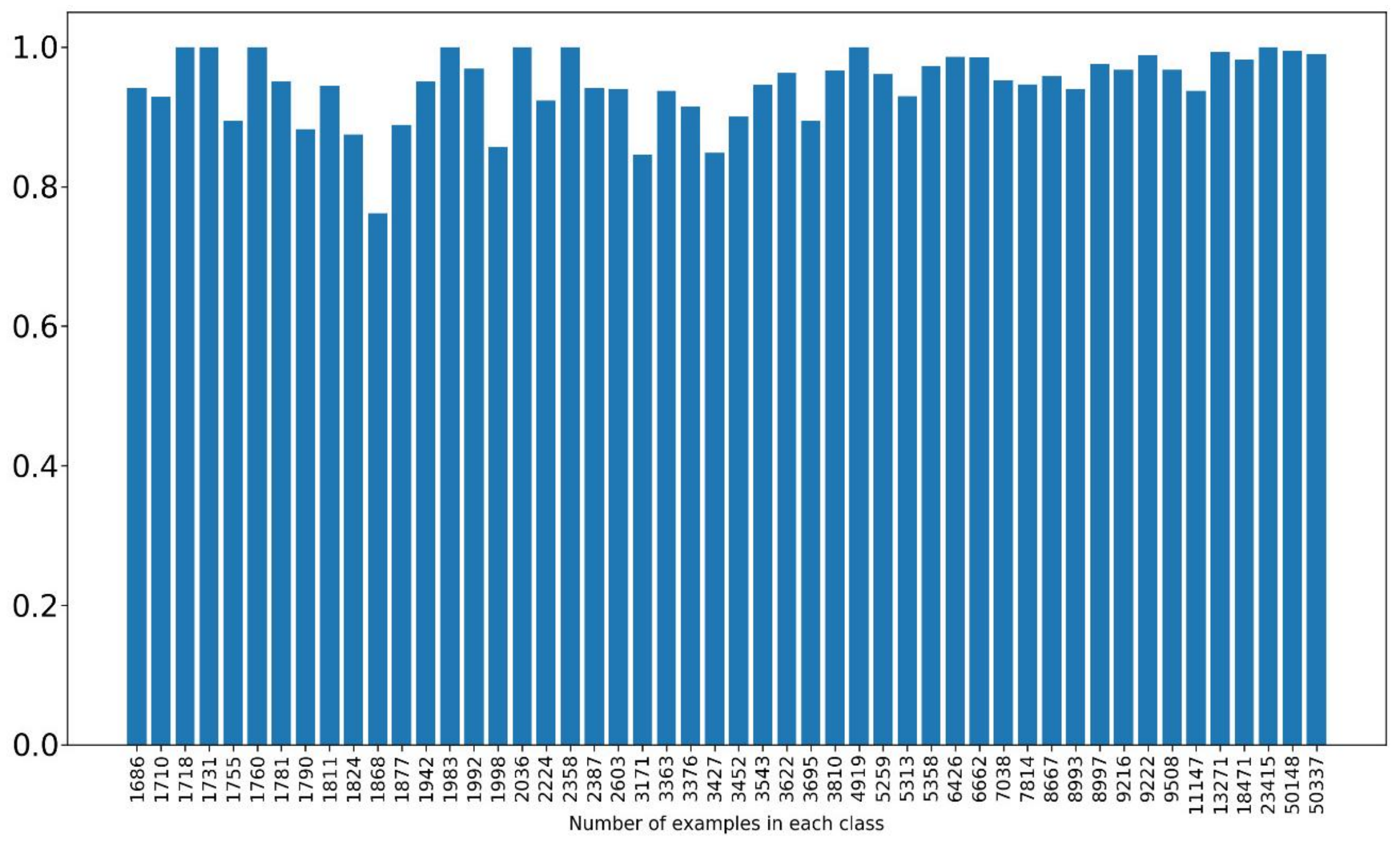

Fig. 4:Accuracy Per Class for ResNet-18 with 50 classes

Understanding and Visualizations

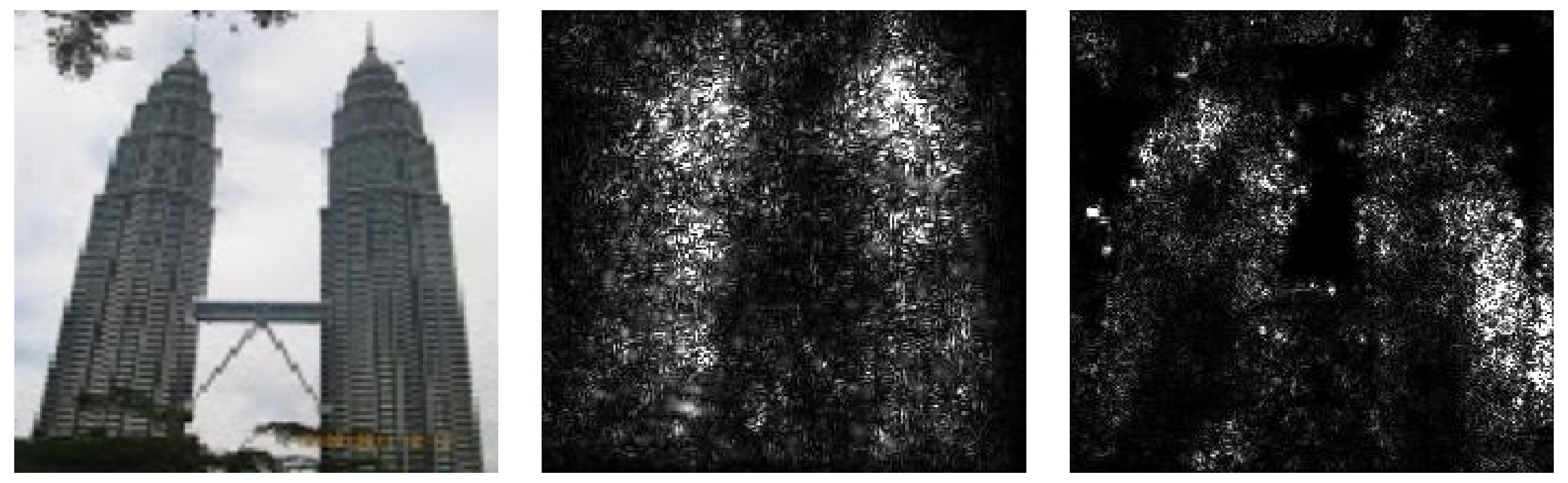

Fig. 5:Saliency Maps on ResNet-18 and VGG-16 [1]

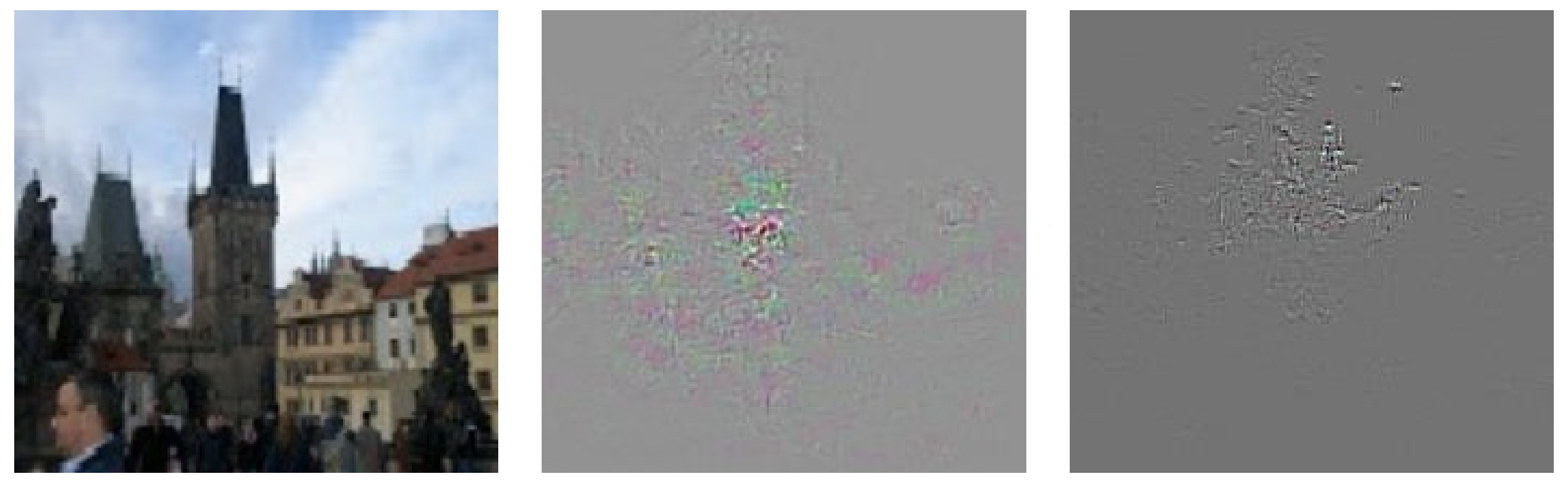

Fig. 6:Vanilla Backpropagation on ResNet-18 and VGG-16

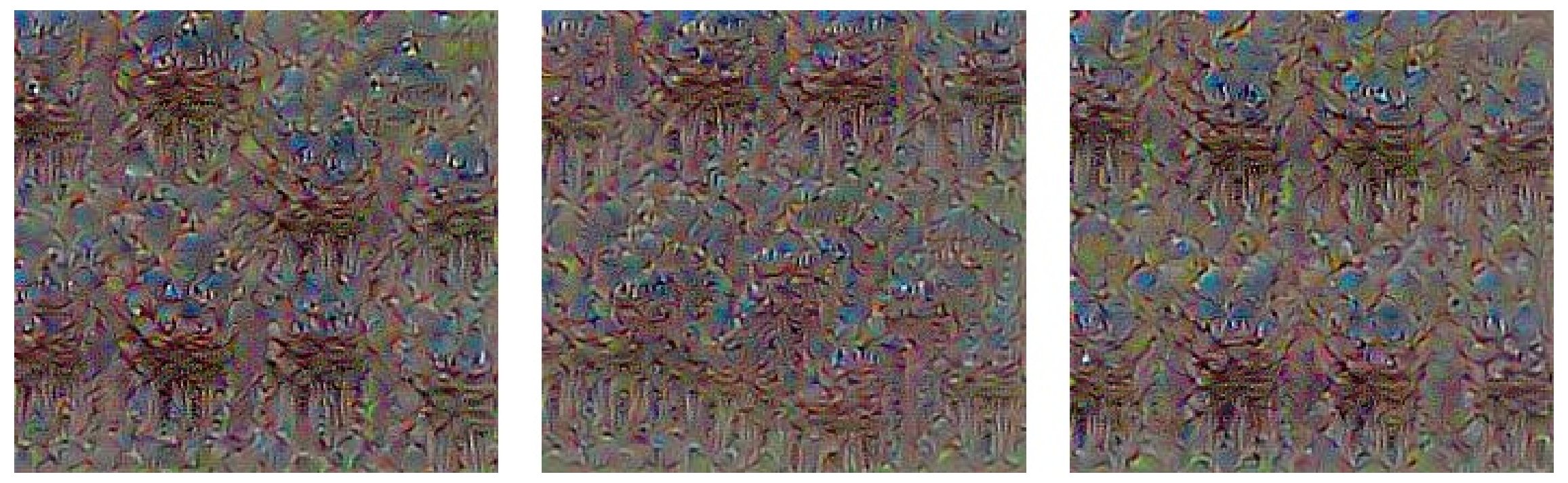

Fig. 7:Class Model Visualization by ResNet-18 [1]

Conclusions

From the first two visualization techniques it seems that the gradients of the ResNet are much more visible compared to VGG gradients. The reason might be in the skipped connections of the Resnet which help the flow of the gradients backward, eliminating the vanishing problem. And despite the highly imbalanced dataset it seems that the ResNet is superior to the VGG no matter the number of classes, without any need for balancing the dataset and still achieving great accuracy.

References

[1] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps.* Visual Geometry Group, University of Oxford, 2014.