R&D Report

Sorting photo   
via landmark recognition and clarity

Martsilenko Ivan, MMAI-1, MSc 1st course

**Abstract.** The problem of photo sorting is considered in the work, the variant of CNN model for landmark recognition and clarity clustering via self-developed algorithm with filters.

# Introduction

A central task is to sort photos via clusters to help user deal with them. That task was presented at CSC Hackathon June 2021 as computer vision task. Me and my team have presented solution that can sort photos via time, location, cleaning up from duplicates, but it still a lot of work to do, such as semantic clustering and filtering low quality images. The idea of ​​the task imposes restrictions on the use of computing power, meaning the use on mobile devices, so we will monitor the execution time of each model. Also need to implement presented idea of image clarity clustering.

Research about semantic clustering has prompted me to try implement landmark recognition classification to sort image from some well-known place to one class for further advice to our user. By the way, presented solution at kaggle competition[1] using a “heavy models” such as Resnet-152, VGG and combination of them. I have decided to try implement faster models such as MobileNet and compare them with ResNet, I think accuracy is not so dominant for that task as evaluation type, so I can suffer it.

Research about clarity analysis didn’t give me any normal results, so In that work I present my realization of image to blured image comparing idea.

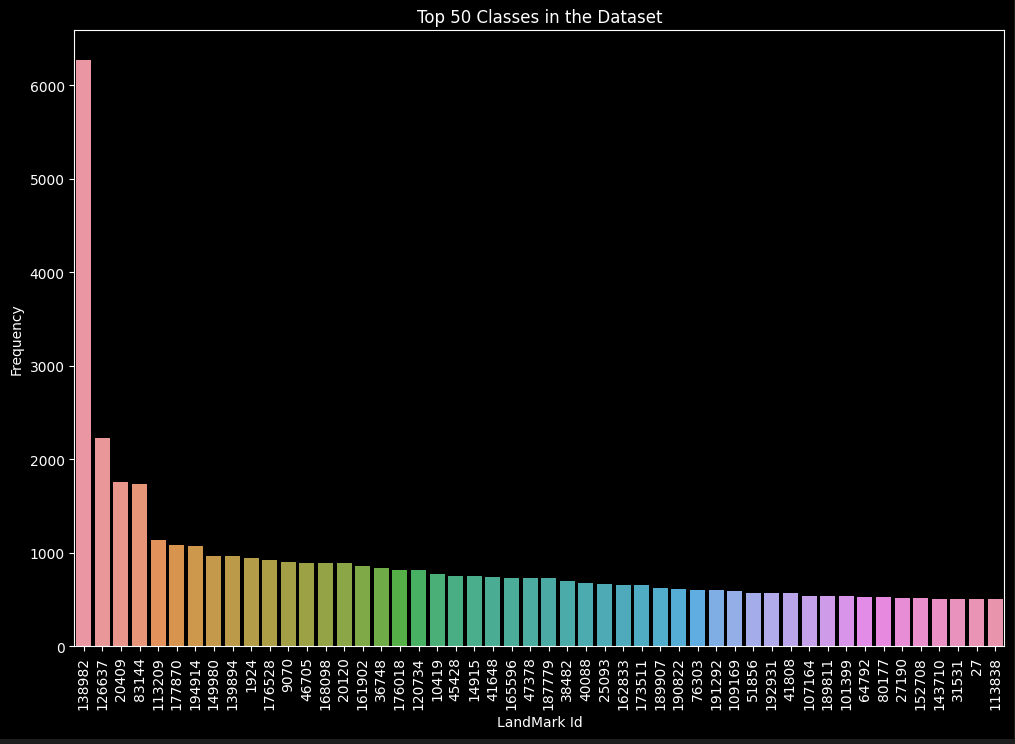
# Google Landmark Recognition

That’s a kaggle competition[1] that provides classifying images by sone well-known places. I have decide to work with 2020 dataset version to have ability to compare my results with previous year competition that is already finished.

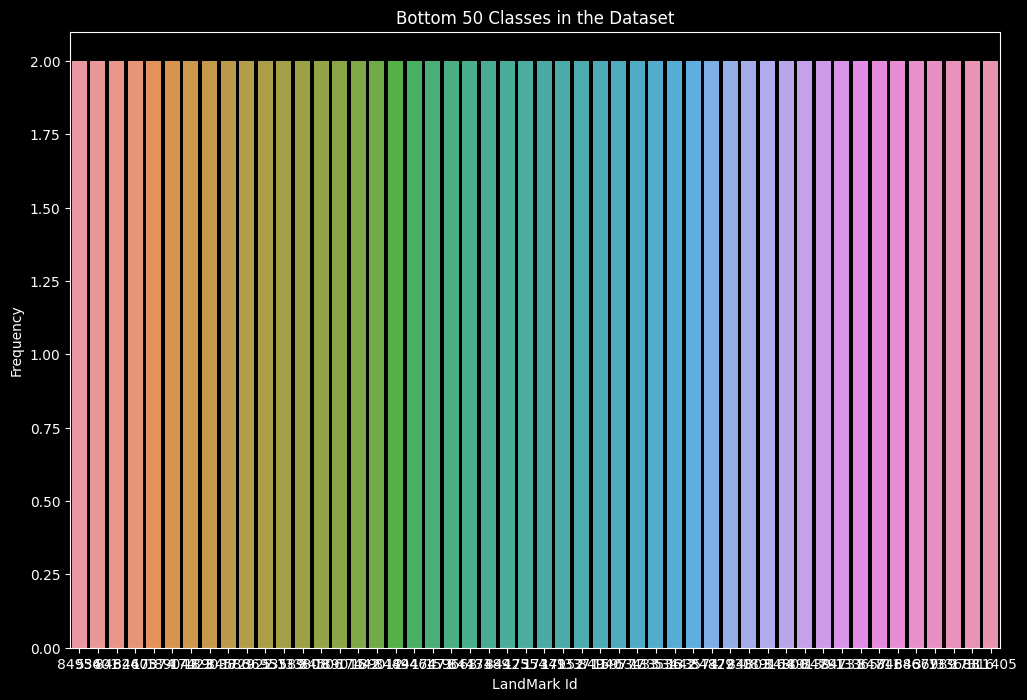
## Exploratory data analysis[a)]

Dataset have very complicated training folder structure and 1580470 images in it. There are 81313 landmarks in the training dataset. I have created dictionary to work with it.

The main things that I found out from clustering analysis is that a hude classes disbalance.



Top classes have thousands of images in it, on the other side is a huge amont of classes that have only a few images in it



Observations from the whole analysis done above:

* There are 81313 unique landmark ids
* There is only one landmark which has more than 6000 images
* Number of images per landmark id ranges from 2 to 6272.

To deal with it I decided to slice a part of dataset for my training I have wrote a script(b) that took only 100 classes to my working dataset, later I will change that dataset to 400 classes. Each class have train, validation and testing part that divides as 120, 40 and 40 images in each.

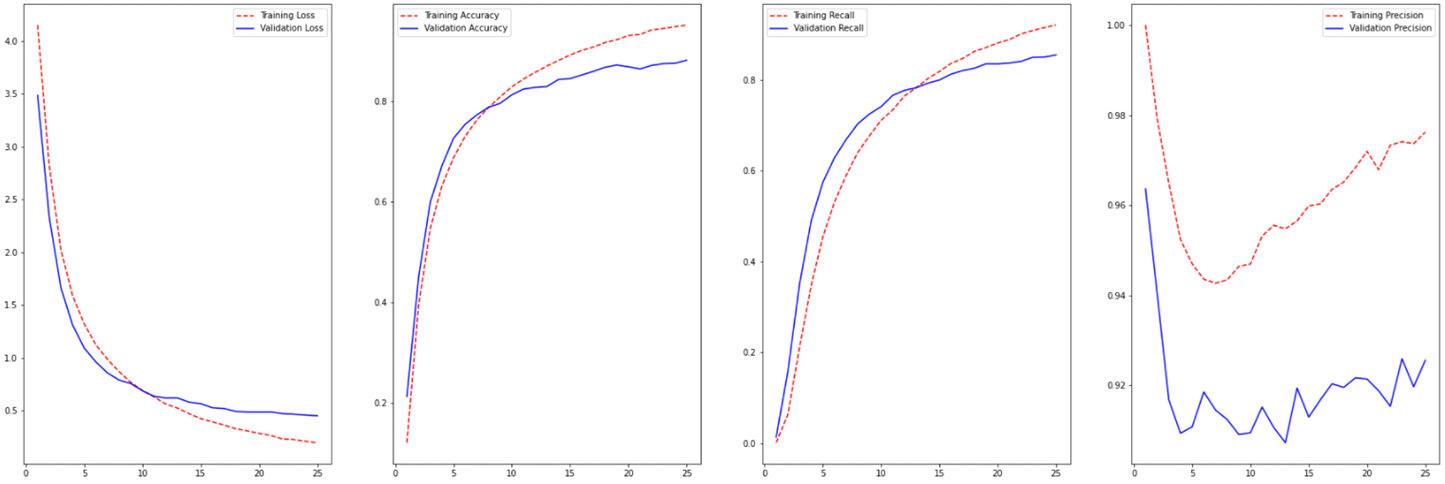
## MobileNet[2,3]

I have started with MobileNet2 CNN model, because it has a good combination of performance and other metrics. But I tuned it for my situation via adding average polling and Dense layer with softmax activation function and 100 (later 400) classes.

Firstly, I have tried to teach it for 25 epochs with such testing results:

loss: 0.46386009454727173, accuracy: 0.8755000233650208

precision: 0.9179844260215759, recall: 0.856249988079071



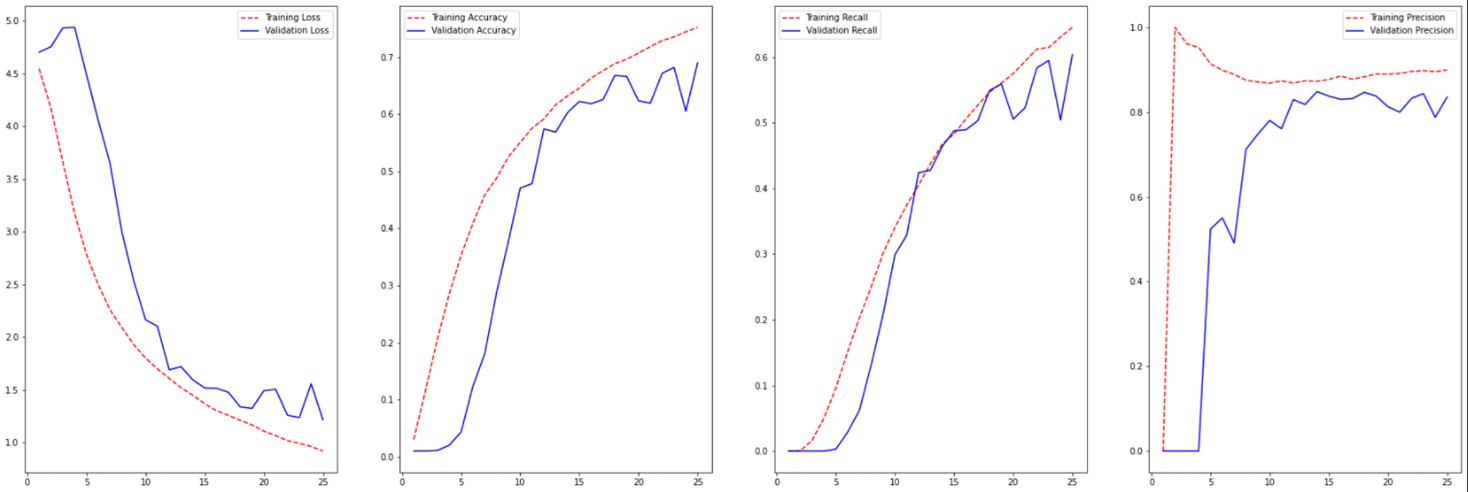
Model have acceptable metrics result and its evaluation time is 35.76 seconds per 16 thousand of images.

After that, I decided to try new – third version of MobileNet CNN.

I started with 25 epochs and got such testing results:

loss: 1.2078657150268555 accuracy: 0.6850000023841858

precision: 0.8452380895614624 recall: 0.6035000085830688

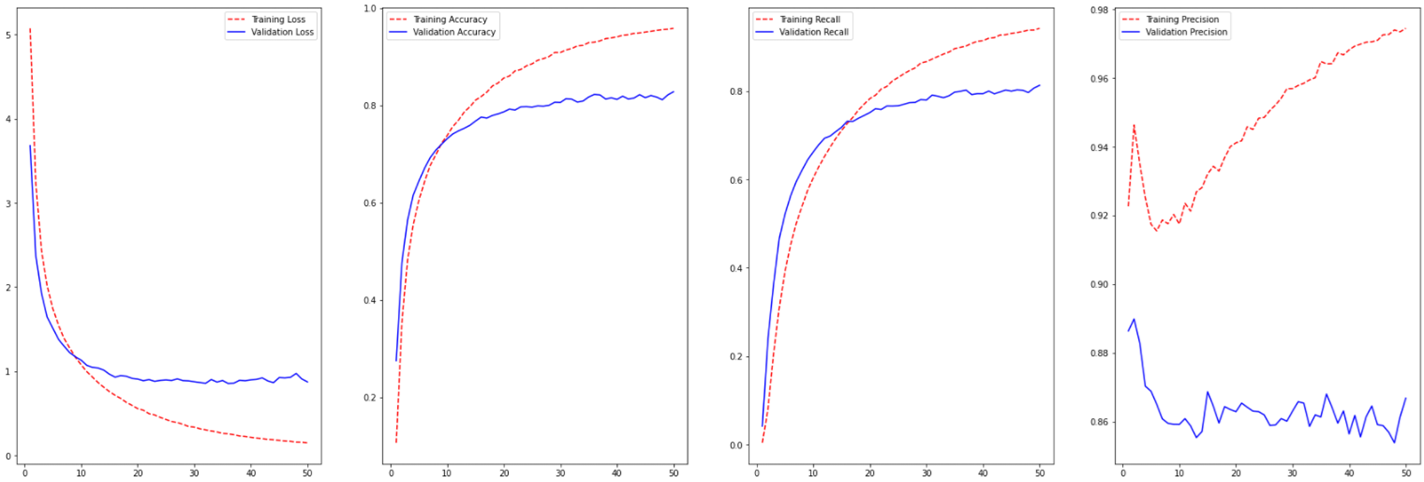


These results are not as good as MobileNetV2, but its evaluation time is almost two time less – only 19:15 seconds per 16 thousand images.

In case of is still big training loss I have decided to train a 25 more epochs for that network, and receive such testing results:

loss: 0.8514732122421265 accuracy: 0.8276875019073486

precision: 0.865370512008667 recall: 0.8123124837875366



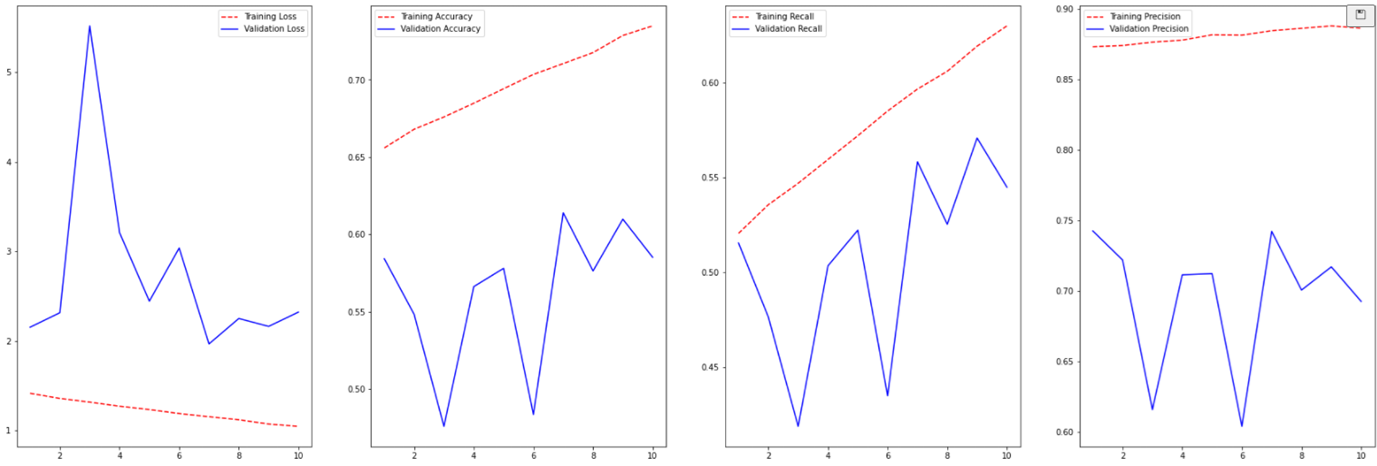
I can’t train MobileNetV3 without overfitting, so I compared it in that state of 50 epochs.

## ResNet[4]

I decided to compare my models with ResNet 50. That Neural Network need much more powerful hardware to train, so I have split that process in a few parts, and don’t have one big plot of training, it’s connected with some numpy and tensorflow checkpoints bug. So after 45 epochs of training the testing results:

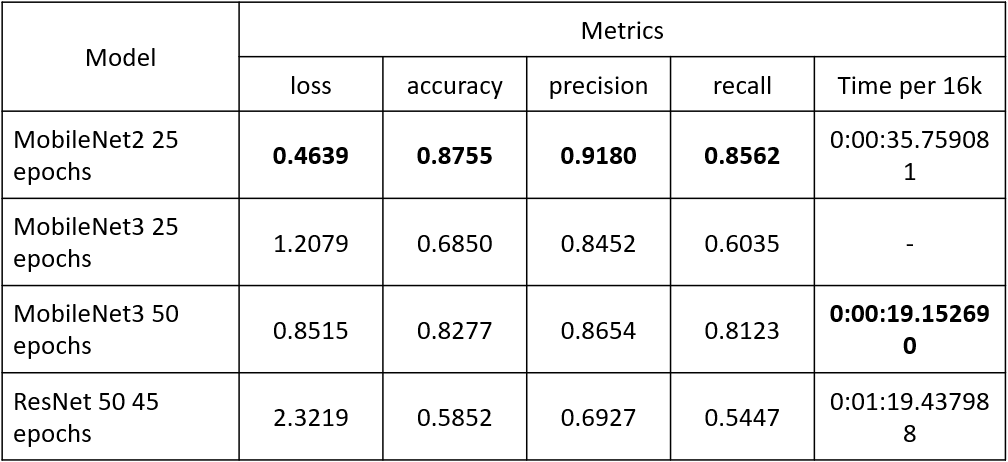
loss: 2.3219 accuracy: 0.5852 precision: 0.6927 recall: 0.5447

Last 10 epochs.



As we can see results are not even close to MobileNet, It can be some bug with data preprocessing or something like that. And its evaluation time is 79,44 second per 16 thousand images.

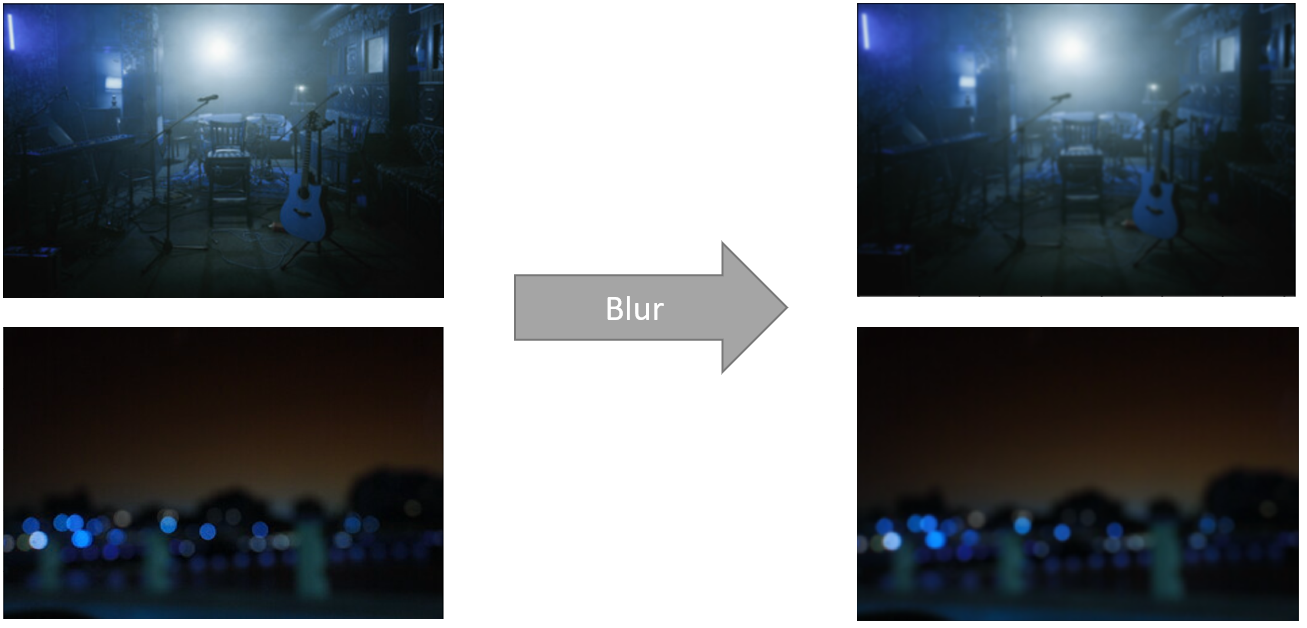
## Comparative analysis



As we can see ResNet is lose for all categories. Even if MobileNetV3 working fuster, I decided to use MobileNetV2 as main model for that app.

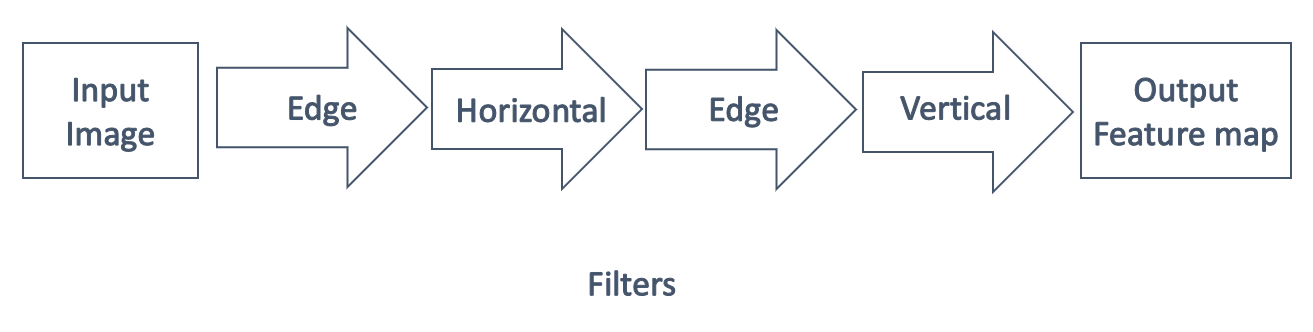
# Clarity clustering[e)]

The main idea of clarity check is to compare how effective blur is. Because it image with bad clarity is already blur.



To compare image with its blur variant model need to extract feature to sum feature map, after that calculating sum of all difference in these activation maps. After all of these actions we can cluster images by that clarity score.

## Feature extractors

Feature extractors is a combination of filters in my case this is edge, horizontal and vertical filters.

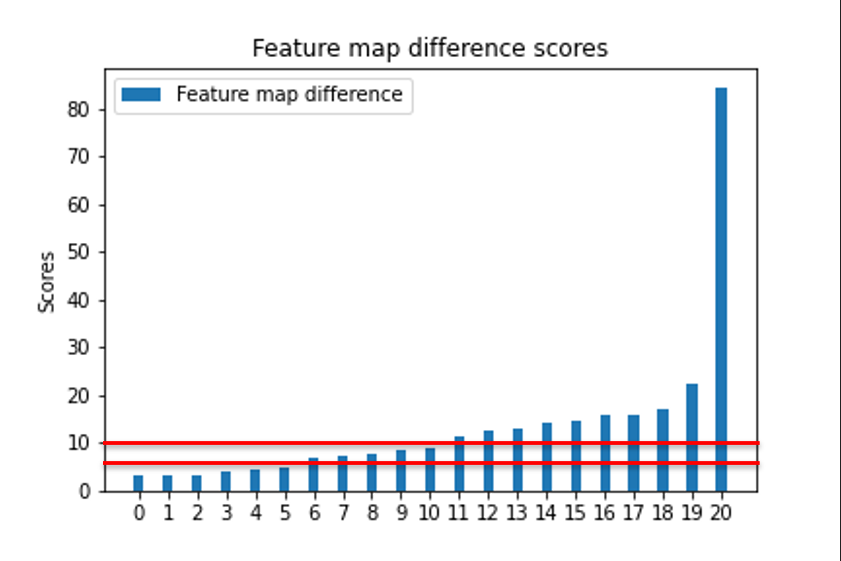
It can have variants and need further research to choose the best option.

## Scoring

To calculate the clarity score we need to calculate sum per each element in both comparing feature map, to do this is much easier to subtract blur image feature map from original image feature map and calculate sum of elements.

## Clustering

After all we have list of scores.



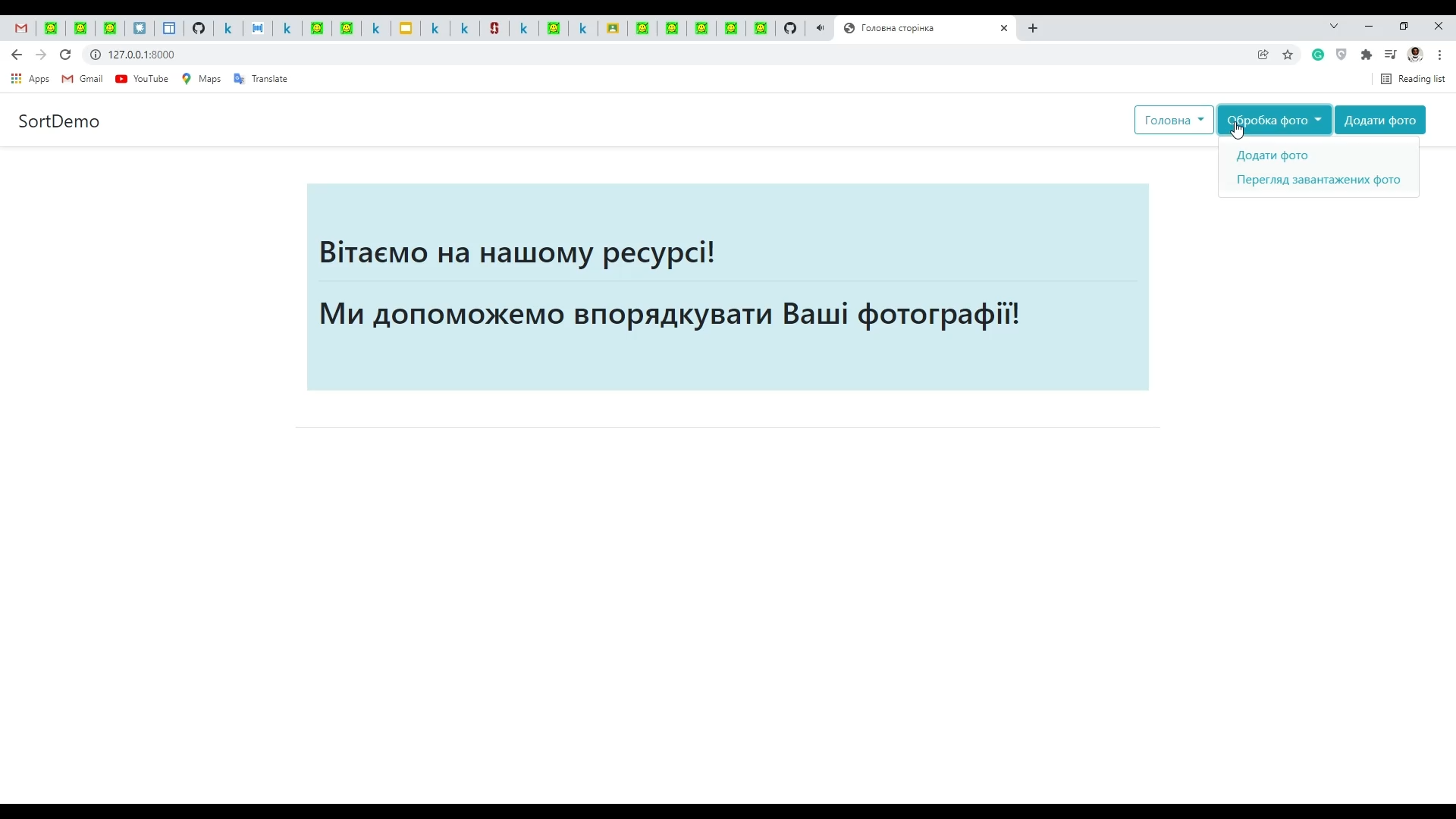
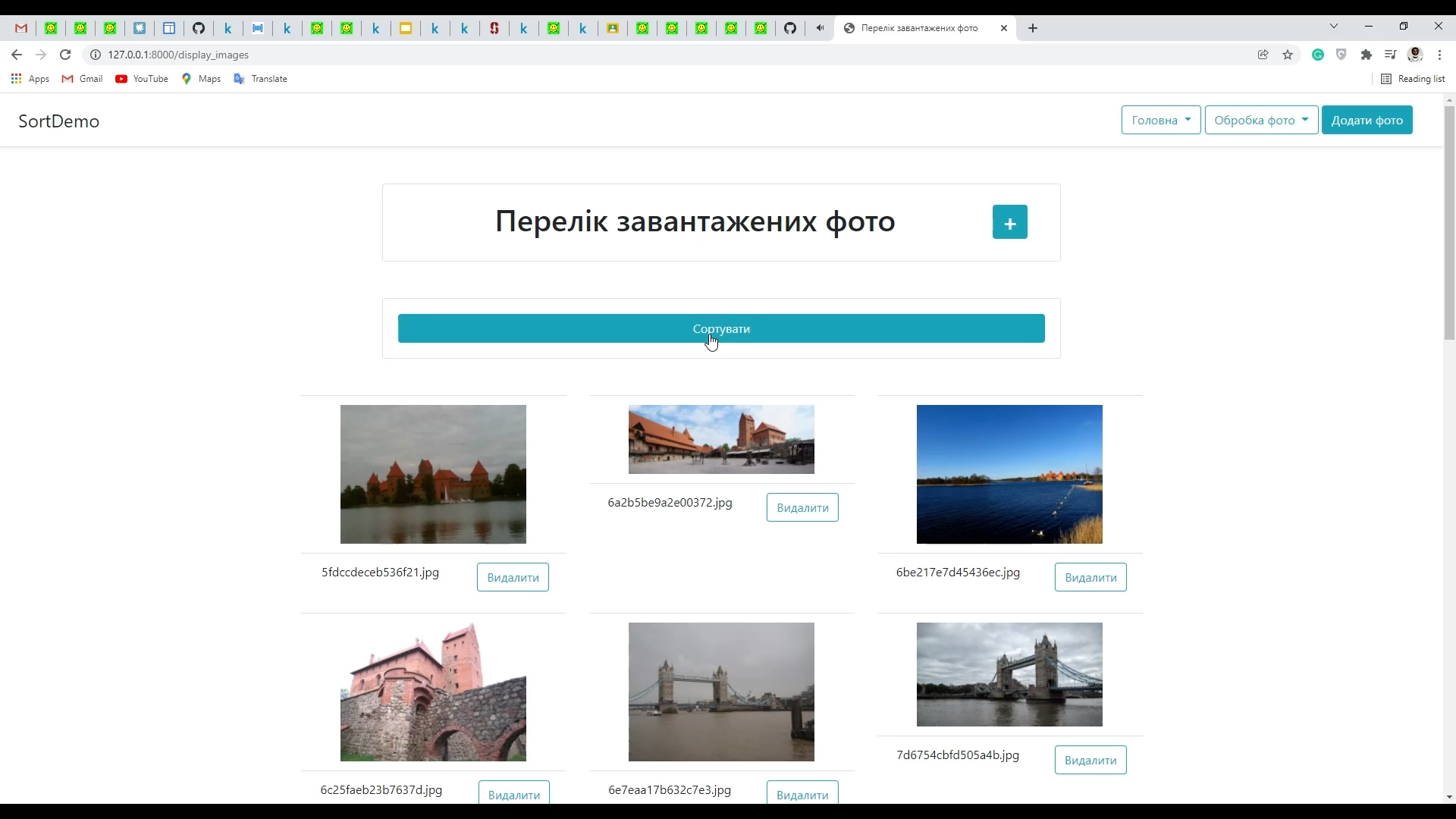
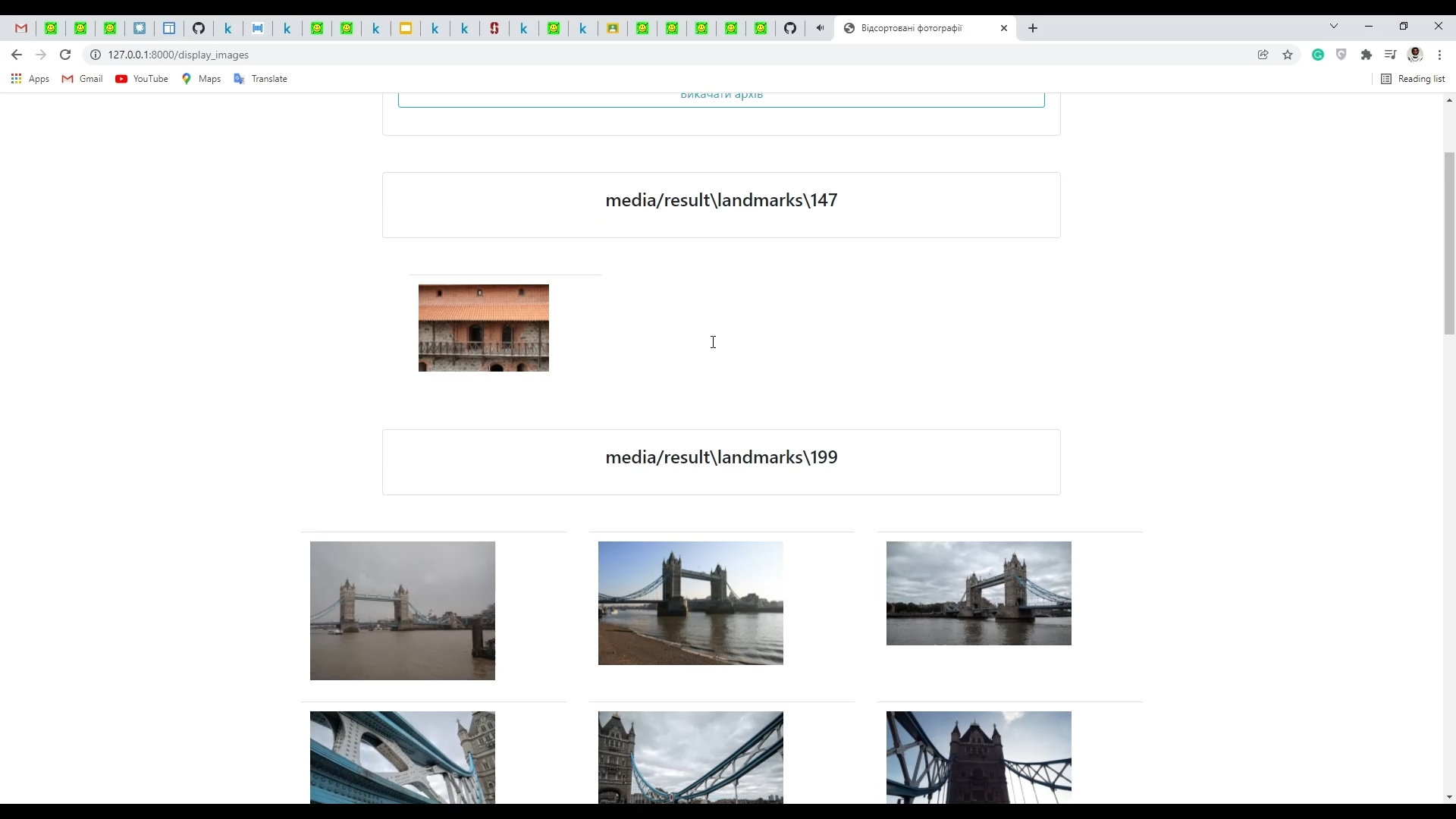
At current stage I divide classes by thresholds, but in further research I will implement classification via density analysis such as DBSCAN, Agglomerative Clustering, OPRTICS, MeanShift or such as K-means.

## Performance

That algorithm has evaluation time 1.2 sec per image right now. But it can be easily optimized and rewrite via c++.

# Demo

In that laboratory assignment I present evaluation of system such as web-site created via python framework Django.

# Conclusion

To sum up, in that paper I have done research about task and presented working demo as website with sorting photos via self-trained MobileNetV2.

Clarity algorithm is still not finished but it works and can be upgrade.

In that work I used in practice my knowledge and skills in machine learning and computer vision, that provides me to deal with that task.

# Sources

1. <https://www.kaggle.com/c/landmark-recognition-2020/code>
2. MobileNetV2: Inverted Residuals and Linear Bottlenecks, A. Howard, M. Sandler and others, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520
3. Searching for MobileNetV3, A. Howard, M. Sandler and others, ICCV 2019
4. Deep Residual Learning for Image Recognition K. He X. Zhang S. Ren – CVPR 2015

# Adds

1. eda.ipynb
2. prepare\_data.ipynb
3. MobileNet.ipynb
4. Resnet.ipynb
5. Clarity.ipynb