Crowd Analysis for Intelligent Surveillance

A PROJECT REPORT

For

INTERNSHIP
by

Jakub Milasz
AGH University of Krakow
Geoinformatics and Applied Computer Science department
Seventh semester bachelor student

Under the supervision
Of
Dr. Neelam Chaplot

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,
SCHOOL OF COMPUTER SCIENCE AND EBGINEERING,
MANIPAL UNIVERSITY JAIPUR,
RAJASTHAN, INDIA-303007
July-August, 2025

Contents

Introduction

These days, we often face a problem of overcrowded or congested areas. This problem commonly occurs in public places or during the various events. The safety decreases due to the large number of people in one place. It becomes easier to pickpocketing or carrying out terrorist attacks. Moreover, crowd aids the spread of infectious diseases and leads to epidemic or pandemic. Therefore, it is very important to create a system which improves security. For this reason, the Deep Learning methods can be applied. It can be suitable approach because these methods are widely used in different cases successfully. The system, based on images from cameras, will provide results from which conclusions can be drawn and security can be improved. One component of the system can be the prediction of the number of people in an image. This will help us determine the current number of people in a given area. It will be possible to estimate suitable threshold above which additional security measures should be introduced. For this purpose, we will compare different methods which can be used for this problem. Then, we will apply Explainable Artificial Intelligence which will help us to understand the operation of the methods in more detail. Finally, we will compare results given by Explainable Artificial Intelligence and draw proper conclusions.

Literature Review

In the literature lots of methods are described for crowd estimation. Here there are some of them.

Aman Ahmed et al. [1] proposed in 2021 approach included two modules with two models. The first one was MobileNet combined with Single Shot Detectors (SSDs) used for identifying, tracking and counting people in the image. For this purpose, Unique Person Detection Algorithm was applied. After extracting the part with person, the age and gender classification was performed. In this module they used VGG16 model. For age classification, they used dataset with 15910 JPG and PNG images divided into three classes: 1-30, 30-60 and 60+. For gender classification, they used dataset with 3966 JPG images divided into two classes: male and female. In both cases the test size was 0.2 and the training size was 0.8. For age classification, an accuracy of 69% was achieved, while for gender classification it was 90%. The authors attributed the classification inaccuracies to the quality of the images.

Yingying Zhang et al. [2] proposed Multi-Column Convolutional Neural Network (MCNN) which estimated crowd using density maps. Each column in the network employs specific filters of different sizes in order to adapt to various heads size in image caused by perspective. They also compared their conception with other methods such as: LBP+RR (Local Binary Pattern + Ridge Regression), MCNN-CCR and another CNN-based method designed to handle variations in scenes for crowd estimation. For training model they used ShanghaiTech dataset with 1198 images divided into part A and part B. For model evaluation and comparison with others, Mean Absolute Error and Mean Square Error were applied. The MCNN model achieved the best results among all methods. For part A, the error values were: MAE = 110.2 and MSE = 173.2, while for part B: MAE = 26.4 and MSE = 41.3. They emphasize that their model can also be applied to other datasets.

Zhiqiang Zhao et al. [3] proposed Cross-Level Dilated Convolutional Neural Network (CL-DCNN). Method is used mainly for crowd counting tasks. The network is based on three key components: the backbone module, Dilated Contextual Modules (DCMs) and a fully connected layer. They use density map and gave the loss function formula. They compared CL-DCNN with other methods using two metrics: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). They tested their approach on various datasets including ShanghaiTech dataset. On this dataset, CL-DCNN got the best MAE on part A which was 52.6. The authors made also some comparison with different number of DCMs. For part A of ShanghaiTech Dataset the best is one, and for part B the best is three. They also checked the Backbones such as AlexNet, VGG16 and ResNet50. VGG16 got the best results on both parts of ShanghaiTech dataset.

Sharath S.V. et al. [4] describe the problem of crowd counting. For this purpose, they used a VGG16 model with transfer learning, pre-trained on ImageNet images. The dataset used for their experiments was the ShanghaiTech dataset, divided into two parts containing crowd images. The authors had to preprocess the dataset into a form suitable for training. This involved image normalization and resizing to 224 × 224 × 3. They also performed data augmentation, including flips, rotation, translation, cropping, and the addition of Gaussian noise. The model architecture consists of 16 layers with ReLU activation. After the convolutional layers, Max Pooling with stride 2 is applied. The number of filters in the layers increases in the order 64, 128, 256, 512 and then decreases in reverse order. The last layer has 1 filter and generates a heatmap. The model was trained on 300 images and tested on 182. Authors set all obligatory parameters and used euclidean distance as the loss function, and MSE as the evaluation metric. The model was trained for 50, 100, and 150 epochs. The best results were obtained with 50 epochs, with 83% training accuracy and 79% validation accuracy.

Methodology

Data was collected from ShanghaiTech dataset as described in Dataset paragraph. Then preprocessing was done and various models such as VGG16, MobileNet, EfficientNetB7, Xception were implemented with the training set and the test set.

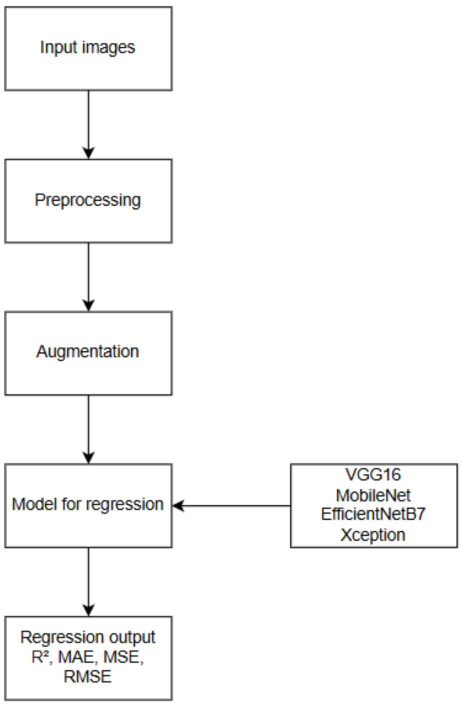

Figure 1: Architecture Diagram for crowd counting

Dataset

The ShanghaiTech dataset was taken from Kaggle. It encompasses 1198 annotated images with a total of 330165 labeled head locations. The dataset contains two parts: Part A (more crowded, 300 training and 182 testing images) and Part B (less crowded, 400 training and 316 testing images). Images from Part A were gathered from the Internet, while images from Part B were gathered on the busy streets of Shanghai. In part A the range of number of people in images is form 33 to 3139 and in part B the range is from 9 to 578. Each part contains also ground_truth directory with location of peoples' heads on the image. For this specific problem only part B was used.

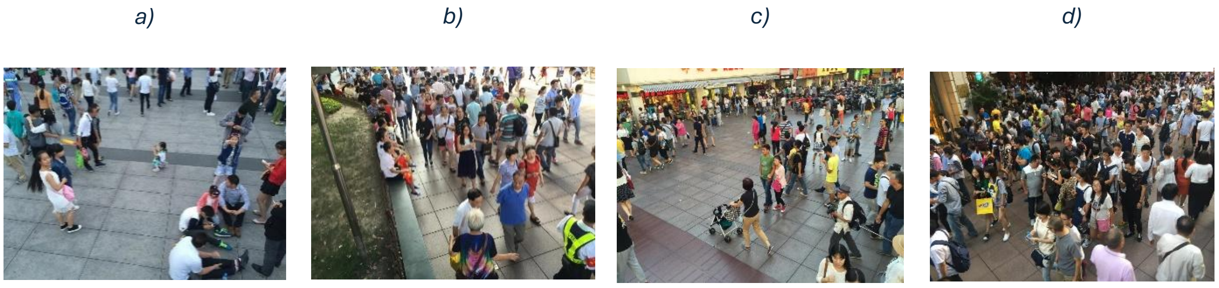

Figure 2: Sample dataset from ShanghaiTech dataset: a) 29 people, b) 90 people, c) 156 people, d) 238 people

Image preprocessing

Before starting the training, it was necessary to prepare the training set containing images adapted to training. Due to different sizes of images, all images had to be rescaled to size 224 × 224 × 3, because most of the transfer learning models were trained on such images. Rescaling with preserving the original length and width using black padding was attempted but it had no influence on final model effectiveness. Therefore, this approach was abandoned due to optimization considerations. During image scaling, the Lanczos resampling was applied in order to maintain appropriate quality. This is the technique for interpolating which offering superior image quality compared to simpler methods like nearest neighbor and bilinear interpolation. Each model has specific requirements related to images such as range of pixels values. Therefore, every model has package in Keras with specific method *preprocess_input*. Every image was assigned the number of people present in it, which was obtained from the corresponding file from ground_truth directory.

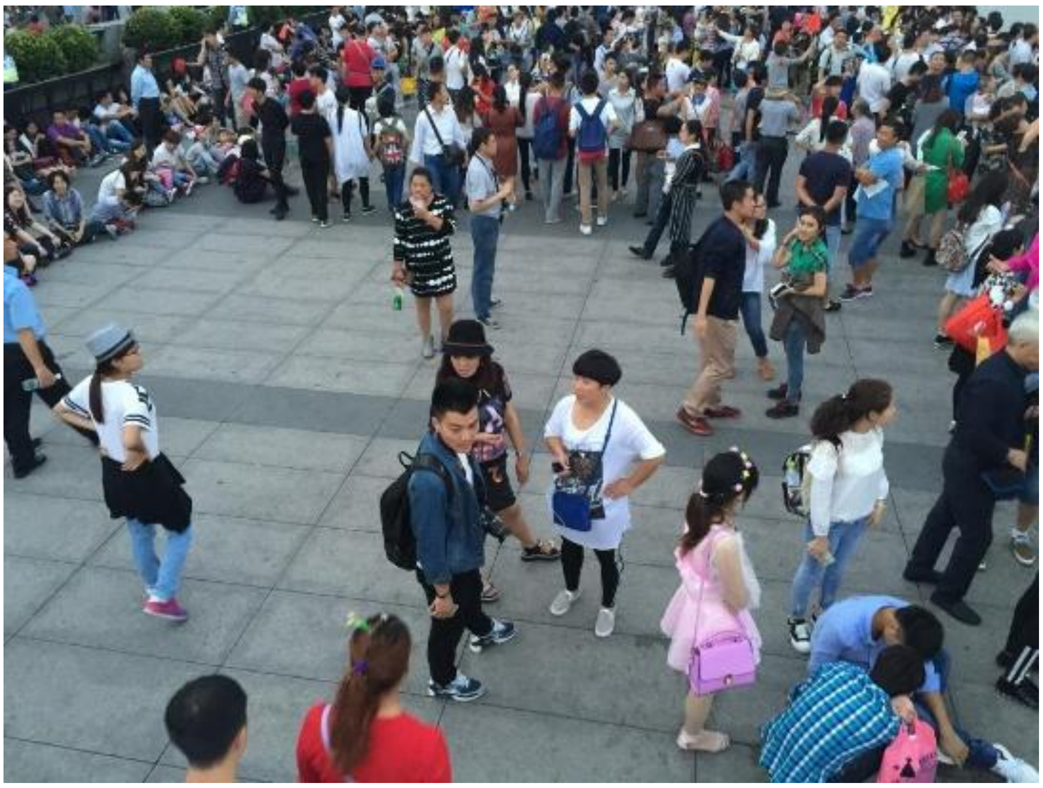

Figure 3.1.: Image before preprocessing.

Figure 3.2.: Image after preprocessing

Data Augmentation

Data augmentation was performed on the training set in order to introduce some variety in data. It consisted of random flip with horizontal direction, random zoom and random rotation, both with a 10% coefficient. Moreover, the order of images in training set is shuffled for each epoch so that the model would not learn from the order of images.

Models

VGG16

This is a convolutional neural network architecture proposed by Visual Geometry Group (VGG) at the University of Oxford. The model is simple and achieves strong performance on various computer vision tasks. The model contains 21 layers: 13 convolutional layers, 3 fully-connected layers and 5 Max Pooling layers. The name comes from 16 weighted layers. Convolutional layers have filters size of 3 × 3 with stride 1 and Max Pooling layers have filters size of 2 × 2 with stride 2. Convolutional layers are divided into 5 internal parts. After each part the Max Pooling is applied. Layers in these parts contain different number of filters:

- Conv-1 layer: 64 filters
- Conv-2 layer: 128 filters

- Conv-3 layer: 256 filters
- Conv-4 layer: 512 filters
- Conv-5 layer: 512 filters

The parameter *include_top* set to *False* excludes 3 fully connected layers.

MobileNet

MobileNet is a simple and effective convolutional neural network used in mobile vision applications. It is applied, among others to object detection and classification. It was published in 2017. The model is based on depth-wise separable convolution blocks which includes depth-wise convolution for applying single filter for each input channel and then point-wise convolution – computes output linear combination from depth-wise using 1 × 1 convolution. It contains in total approximately 90 layers, where 28 are convolutional and the rest is related to batch normalization or padding. The convolutional layers specification is as follows:

- One standard convolution layer with size of 3 × 3 filter, stride 2, 3 input channels and 32 output channels
- Depth-wise convolution– size of 3 × 3 filters
- Point-wise convolution – size of 1 × 1 filters

After that, the Global Average Pooling is applied, but in this case *include_top* parameter is set as *False* and because of that GAP and further layers are not applied in execution.

EfficientNetB7

This is convolutional model (ConvNet). The main objective of the method is to scale all dimensions of depth, width and resolution using a simple and effective compound coefficient. Model includes total 813 layers. It is based on 7 blocks with MBConv (Mobile Inverted Bottleneck Convolution). Within the block there are:

- channel expansion with size of 1 × 1 convolution
- size of 3 × 3 and 5 × 5 filters
- squeeze-and-excitation
- projection to fewer channels using size of 1 × 1 convolution

At the end there is a GlobalAveragePooling2D, but it and following layers as well will be excluded because of *include_top* parameter.

Xception

This is an another computer vision model. It is a successor of Inception architecture. The model was evolved by Francois Chollet at Google, who continued success of the Inception architecture and further perfected it. Xception replaces inception modules with depth-wise separable convolution layers. This model contains approximately 126 layers. It has 3 main components:

- Entry Flow - it convolutional layer used with 32 different filters size of 3 × 3 and stride 2. This is followed by another convolutional layer with 64 filters size of 3 × 3, activated with ReLU. Then, the modified depthwise separable convolution layer is applied, combined with the 1 × 1 convolution layer and stride 2. Finally, the size of the feature map is reduced by Max Pooling size of 3 × 3 with stride 2.
- Middle Flow – depth-wise separable convolution with 728 filters with size of 3 × 3 kernel and activated with ReLU. The block is repeated 8 times.
- Exit Flow - separable convolution which contains 728, 1024, 1536, and 2048 filters, all using 3 × 3 kernels, to extract increasingly complex features.

After that Global Average Pooling is applied, but here *include_top* is *False*, so GAP and following layers are not implemented.

All models are built using a specific base architecture derived from transfer learning, with each base model pre-trained on the ImageNet dataset. The top layers for classification are excluded. Therefore, the architecture of each model is as follows:

- **Base Model** (pre-trained on ImageNet, used as a feature extractor)
- **Flatten Layer**
- **Dropout Layer** (rate = 0.3)
- **Dense Layer** with 128 units and ReLU activation
- **Dropout Layer** (rate = 0.3)
- **Dense Layer** with 64 units and ReLU activation
- **Output Layer**: Dense layer with 1 unit and linear activation

It was also tested how results differ if we unfreeze some layers of base model. For VGG16 and Xception 3 layers were unfrozen and for MobileNet and EfficientNetB7 – 30 layers. To ensure reproducibility of the results, the random seed was set to 42 for the TensorFlow, NumPy, and random libraries in all models.

Metrices

In this section the model evaluation is processed. For this purpose, there were: Mean Squared Error, Mean Absolute Error, R squared and Root Mean Square Error are used.

Explainable AI

Explainable Artificial Intelligence (XAI) was used in order to see, how models make decisions. This term refers to a set of processes and methods that enable humans to interpret and trust the decisions made by AI systems. It makes the „black box" of AI more understable and transparent. For this purpose, the Gradient-weighted Class Activation Map (Grad-CAM) method was used because of its specificity and compatibility with the models. For research we take the last layer before the last layer from our Base Model architecture. Firstly a grad model that outputs a feature map and the model's final prediction is built. Then the loss for Grad-CAM is computed, using a single scalar since our problem is regression. Next, the gradient between the loss and the feature map is calculated. After that, the gradients over the spatial dimensions are aggregated and then a weighted sum of the channels are computed to generate the heatmap. Finally, negative values are set to 0 and the heatmap is normalized. The main objective of the models is to detect people. Therefore, it was checked which areas of the image the last layer of the model before Flatten considers the most (highest density) and which it considers less significant (lowest density) to predict the result. In the figure below, one can observe how the image is interpreted by different models.

Results

Results of models

All results are shown in the tables below.

Model	**MSE**	**MAE**	**R2**	**RMSE**
VGG16	**3779**	40	**0.58**	**61**
MobileNet	4294	**38**	0.53	66
EfficientNetB7	4832	42	0.47	70
Xception	6436	49	0.29	80

Figure 4.1.: Evaluation: all layers frozen.

Model	**MSE**	**MAE**	**R2**	**RMSE**
VGG16	**2140**	30	**0.76**	**46**
MobileNet	2381	**29**	0.74	49
EfficientNetB7	4123	40	0.54	64
Xception	3626	43	0.60	60

Figure 4.2.: Evaluation: some layers unfrozen.

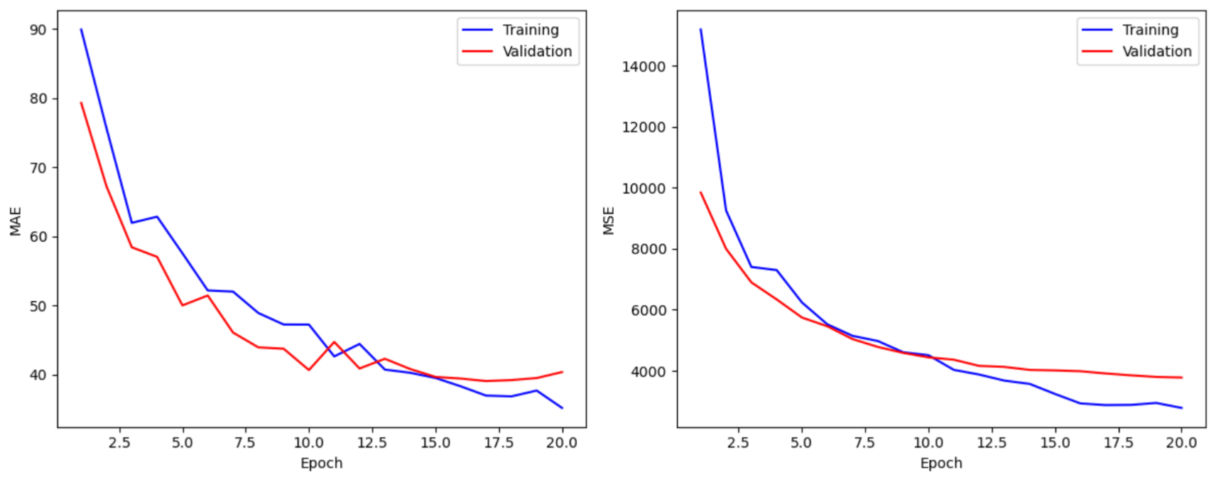

Figure 5.1.: VGG16 training curves with all pre-trained layers frozen.

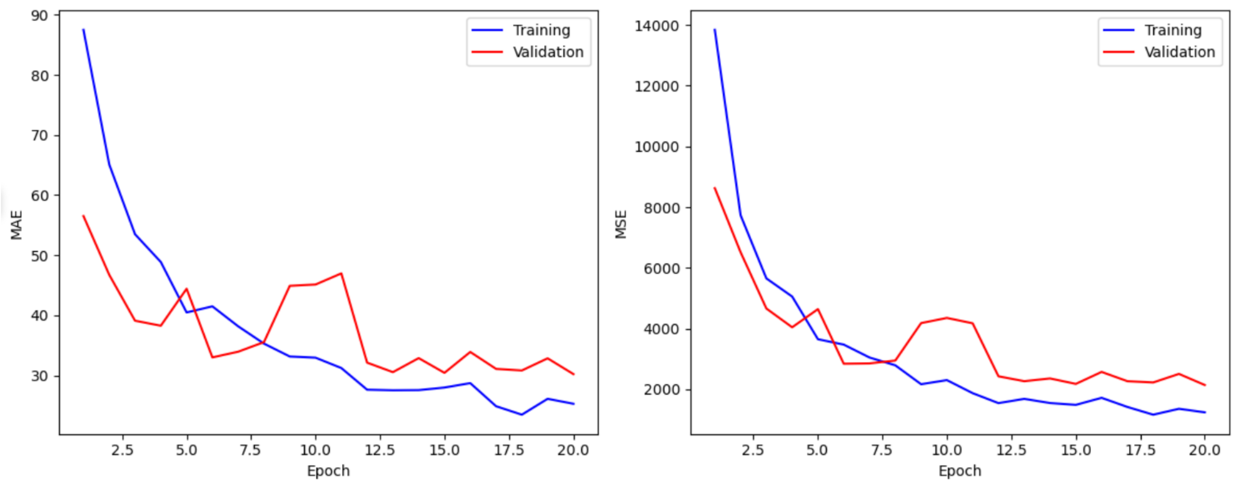

Figure 5.2.: VGG16 training curves with some pre-trained layers unfrozen.

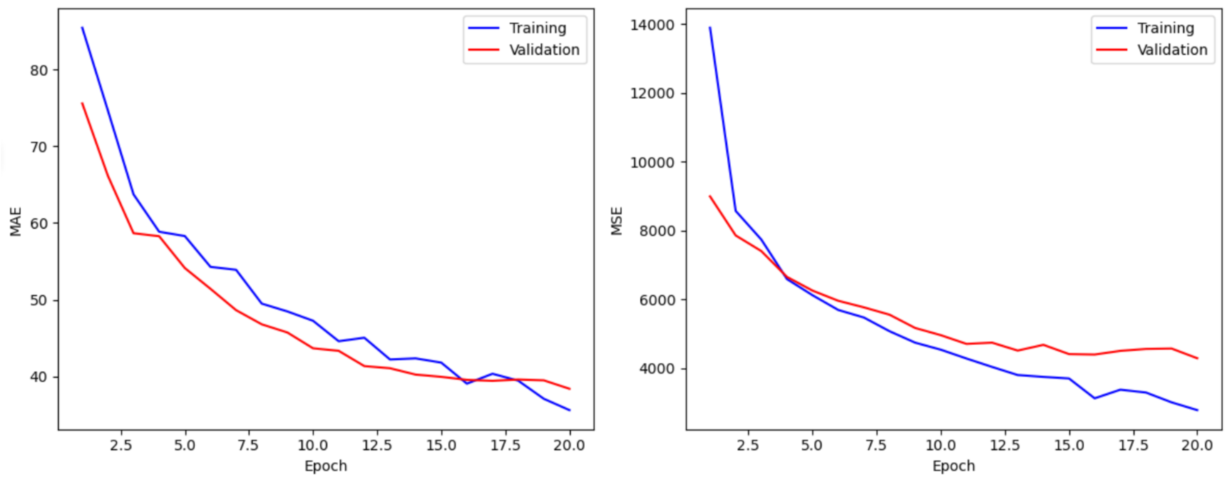

Figure 5.3.: MobileNet training curves with all pre-trained layers frozen.

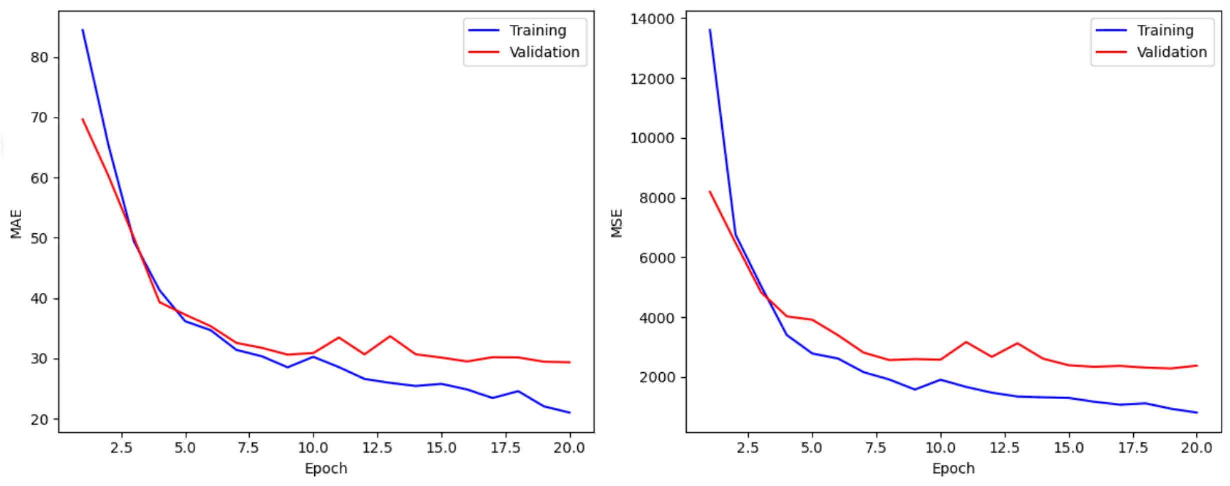

Figure 5.4.: MobileNet training curves with some pre-trained layers unfrozen.

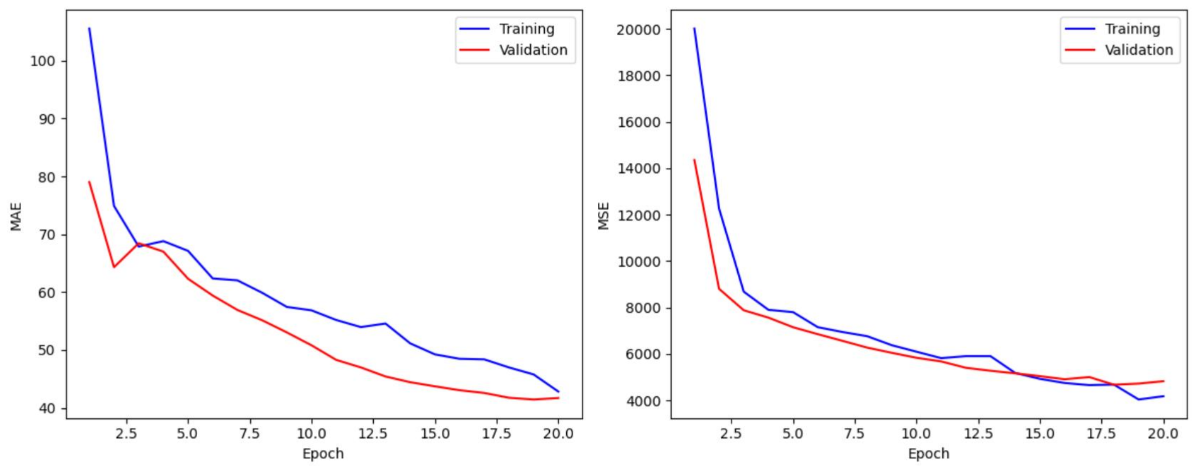

Figure 5.5.: EfficientNetB7 training curves with all pre-trained layers frozen.

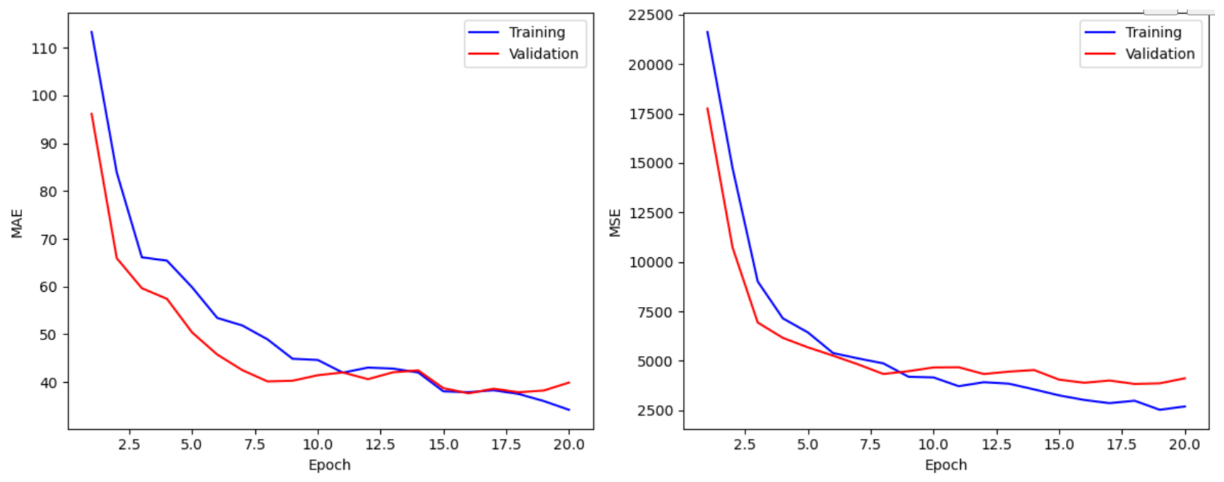

Figure 5.6.: EfficientNetB7 training curves with some pre-trained layers unfrozen.

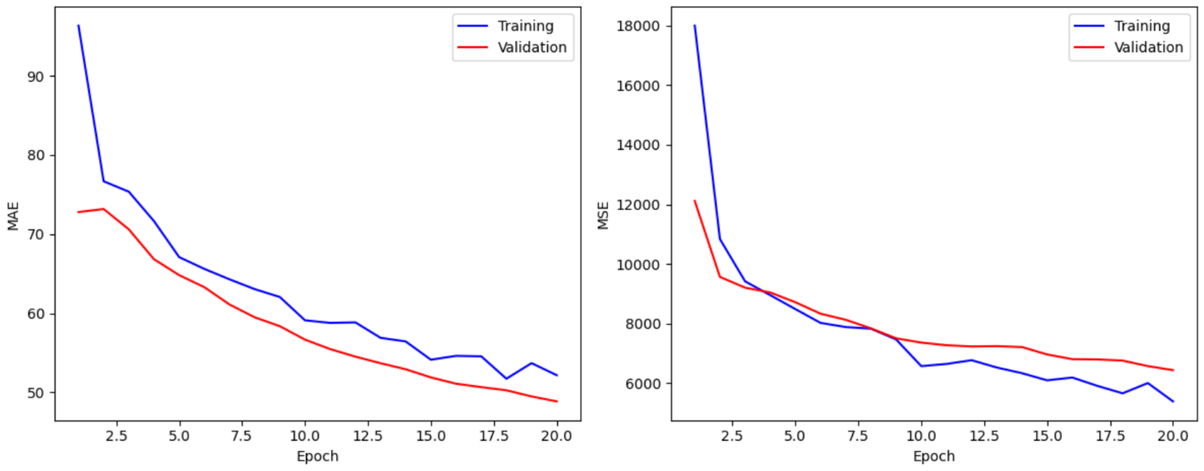

Figure 5.7.: Xception training curves for MAE error and MSE error with all pre-trained layers frozen.

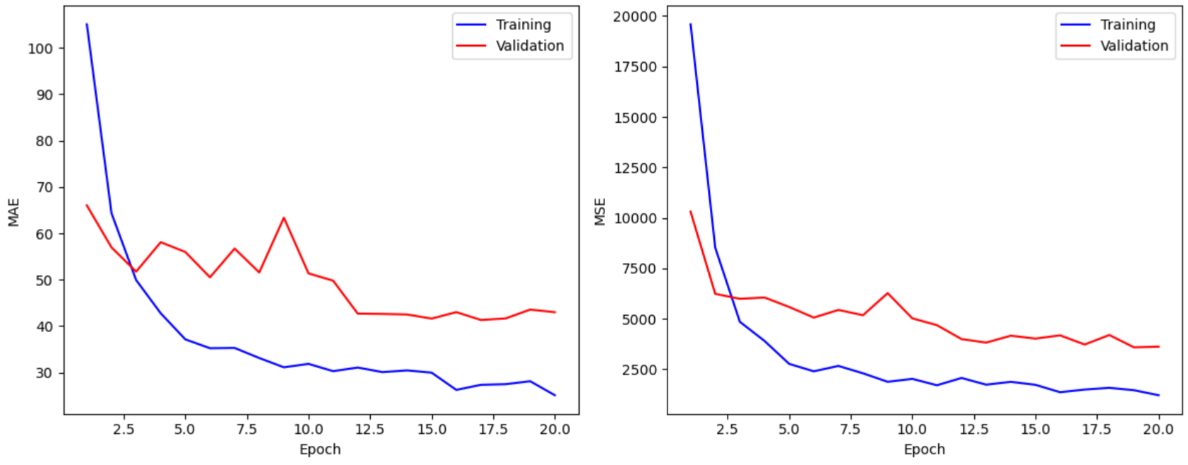

Figure 5.8.: Xception training curves with some pre-trained layers unfrozen.

Results of Explainable AI

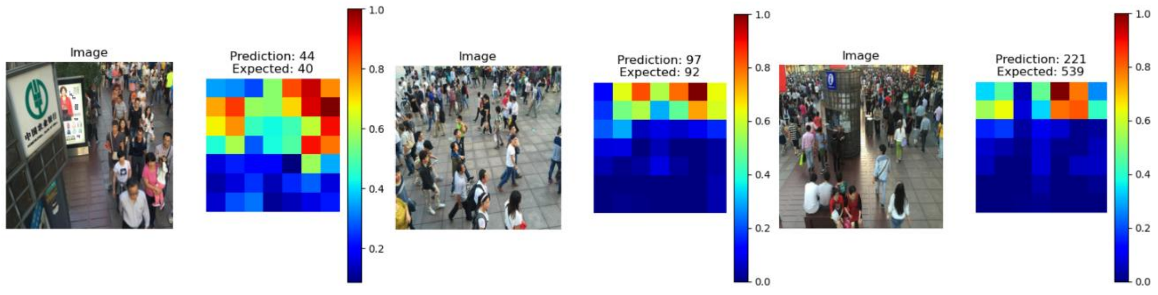

Figure 6.1.: VGG16 maps before the Flatten

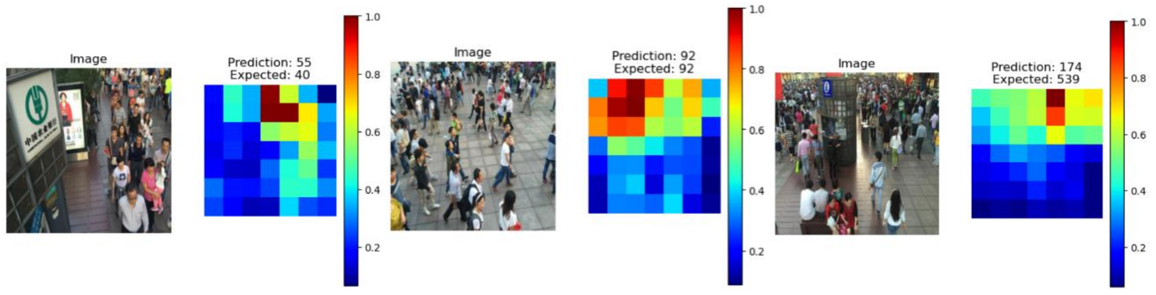

Figure 6.2.: MobileNet maps before the Flatten

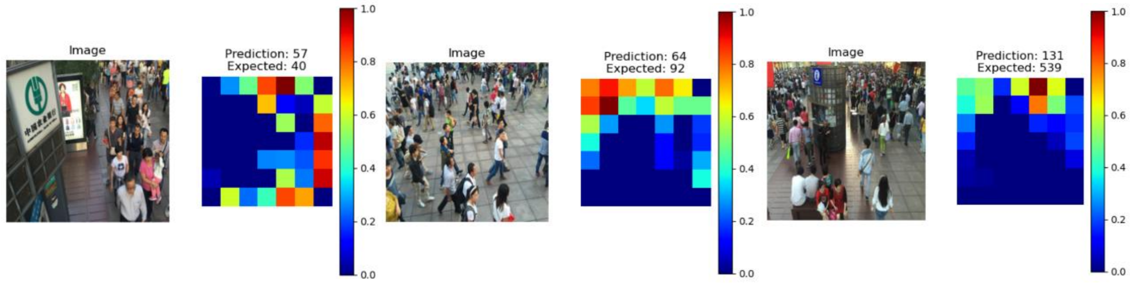

Figure 6.3: EfficientNetB7 maps before the Flatten

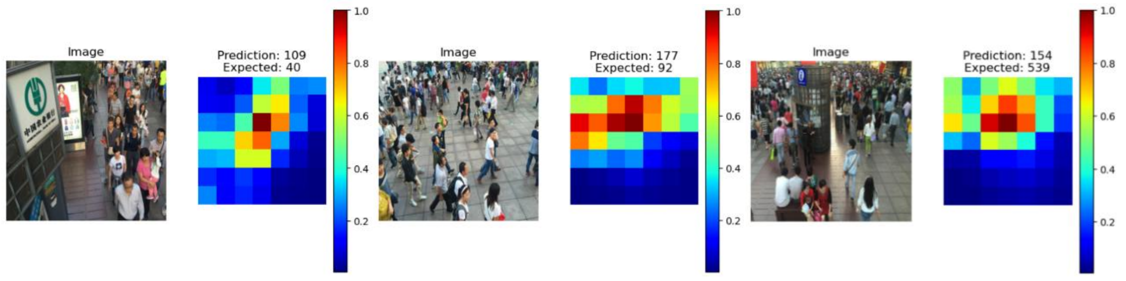

Figure 6.4.: Xception maps before the Flatten

Conclusion

In conclusion, prediction of the number of people can be done in different ways, but the choice of the best method is significant to obtain the most accurate results. Various variants should be tested and it should be remembered that depending on the specificity of the dataset, the effectiveness of individual methods may vary. In the XAI figures, it can be observed that different models consider different areas of the images as the most important. Some handle a given example better, while others perform worse. However, each model had a similar issue with an image containing 539 people. The predicted number was always significantly lower. This could have been influenced by the perspective where the heads of people in the background appear smaller than those in the foreground. For the solution, the MCNN model could be applied which was described in [2]. The best performance on this example was achieved by VGG16 which detected properly the area with crowd on the right-hand side as opposed to the rest of the models. On the other side, it failed to capture crowded area on the left-hand side. It can be also noted that MobileNet predicted the exact number of people in one of examples. Analyzing its heatmap, we can see that it correctly identified the most densely populated areas, whereas other models struggled with this. There were also some cases where models considered non-human objects as crowd.

References

[1] A. Ahmed, P. Bansal, A. Khan and N. Purohit, "Crowd Detection and Analysis for Surveillance Videos using Deep Learning," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1-7, doi: 10.1109/ICESC51422.2021.9532683. keywords: {Deep learning;COVID-19;Pandemics;Communication systems;Focusing;Aerospace electronics;Video surveillance;Deep Learning;Crowd Density Estimation;CNN;MobileNets;Neural networks},

[2] Y. Zhang, D. Zhou, S. Chen, S. Gao and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 589-597, doi: 10.1109/CVPR.2016.70. keywords: {Head;Feature extraction;Neural networks;Image segmentation;Distortion;Detectors;Image resolution},

[3] Zhao Z, Ma P, Jia M, Wang X, Hei X. A Dilated Convolutional Neural Network for Cross-Layers of Contextual Information for Congested Crowd Counting. Sensors (Basel). 2024 Mar 12;24(6):1816. doi: 10.3390/s24061816. PMID: 38544079; PMCID: PMC10975759.

[4] S. V. Sharath, V. Biradar, M. S. Prajwal and B. Ashwini, "Crowd Counting in High Dense Images using Deep Convolutional Neural Network," 2021 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Nitte, India, 2021, pp. 30-34, doi: 10.1109/DISCOVER52564.2021.9663716. keywords: {Training;Visualization;Computational modeling;Transfer learning;Very large scale integration;Convolutional neural networks;Integrated circuit modeling;Deep convolutional neural network;crowd counting;VGG 16;transfer learning;image augmentation},