Crowd Density Estimation Using Deep Learning For Prevention Of Crowd Crush Disaster

HaLim Jun
hjun43@gatech.edu

Abstract **-** This project aims to develop a crowd density estimation model to prevent crowd crush disasters. Concepts of Gaussian filter, autoencoder, and semantic segmentation are utilized. The semantic segmentation model showed meaningful results with recall of 0.71.

1. PROBLEM STATEMENT

A crowd crush is a disaster where an extremely dense population ends up stumbling into each other, causing severe injuries or deaths on a large scale. On 2022 October 1st, a fatal crush occured in East Java, Indonesia, during a football match, causing more than 135 deaths. Within the same month, on 2022 October 30th, a day before Halloween, a deadly disaster of crowd crush occurred in South Korea's popular destination, Iteawon, killing 158 people. Such incidents raised awareness on a weak police system for controlling the crowd, alerting the need to improve the efficiency of police surveillance in case of a mass event. In such cases, an automated evaluation of the crowd's surveillance images can greatly increase the effectiveness of a limited police force

Under the purpose of developing a crowd density estimation system to further prevent crowd crush, this project suggests using deep learning models to develop crowd density estimation of image data.

This paper suggests the model in the following order. In section 2, the dataset used for the modeling is introduced. In section 3, the objective of the model in terms of real-world adaptation is illustrated. In section 4, mathematical and statistical concepts used in the model are described. In section 5, each modeling methodology is explained and in section 6, the model evaluations and the output of the models are suggested.

2. DATA SOURCE

Utilized data set is ShanghaiTech crowd detection dataset[1]. This dataset has original images of crowds and processed images with head marked. It has a total of 1198 images. Considering the computational power, around 500 images were actually deployed in training and testing the model. (300 for training, 200 for testing)

[1] https://www.kaggle.com/datasets/tthien/shanghaitech

Photo1. Original images

Photo2. Original images with head annotation

3. OBJECTIVE

This paper aims to develop a deep-learning model that can calculate the population density from crowd images, in order to signal possible crowd crush incidents and reduce victims to disasters. It selects a convolutional neural network model due to its performance in dealing with image data. Such a model is meaningful in that it can easily be deployed in the real world, due to the abundance and accessibility of public surveillance image/video data. Especially surveillance camera is often launched in where are critical in terms of security and overcrowdedness. For example, anyone can easily access to the real-time webcam video of Vatican, Saint Peter's Square through the internet.(Photo1) Also, real-time video of Temple Bar, Dublin is available, a popular tourist destination where massive numbers of people are likely to gather. Thus, such a model to detect overcrowdedness with images could be easily adapted to real-world crowd detection problems and possibly reduce crowd crush incidents.

Photo2. Realtime Public Surveillance video of Saint Peter's Square, Vatican[2]

Photo3. Realtime Public Surveillance video of Temple Bar, Dublin[3]

[2] Worldcams, https://worldcams.tv/vatican/vatican/st-peters-square
[3] Worldcams, https://worldcams.tv/ireland/dublin/temple-bar

4. CONCEPTS

In this section, statistical and mathematical concepts that have been utilized in the project will be introduced.

4.1 Gaussian Filter

A Gaussian filter is linear filter that smoothes the distribution using the shape of the Gaussian function. A Gaussian kernel is typically used for denoising or blurring images or videos. It has kernels that act as center points for data transformation. The equation for Gaussian filter $G(x)$ is given as below

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where the x is the distance between the kernel and the given point and σ is a hyperparameter that controls the degree of data transformation. In image data, the Gaussian kernel blurs the image more as the σ gets larger.

4.2 Convolutional Neural Network Model

Convolutional neural network (CNN) is a form of neural network that is specialized in processing grid-like data. A kernel, which is a matrix smaller than the image, slides through each pixel, and its multiplication with the overlapping image is saved as an activation map. This process is called a convolutional operation. CNN consists of multiple kernels and an image is processed through convolutional operations with the kernel layers.

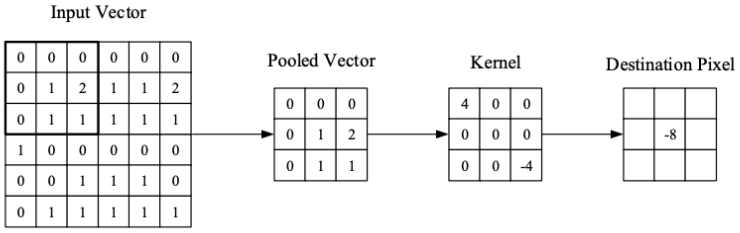


Figure 1. Convolutional Operations[4]

The values in the kernels are parameters of the CNN model and it is optimized with a process called "*Backpropagation*". When the squared difference between the output of the model and the

[4] Keiron O'Shea, Ryan Nash, *An Introdiction to Convolutional Neural Networks, arXiv:1511.08458v2 2 Dec 2015*

ground truth, or loss, is z, we update the model weights w to minimize z. When using stochastic gradient descent, we update the weight using the following rule:

$$\boldsymbol{w}^i \longleftarrow \boldsymbol{w}^i - \eta \frac{\partial z}{\partial \boldsymbol{w}^i}$$

where η is a fixed step size. Backpropagation is a process of comparing the output to desired output, and adjusting the connection weights using the chain rule.

4.3 AutoEncoder

AutoEncoder is a neural network that encodes the input as a meaningful and compressed representation and then decodes it again similar to the output[5]. It consists of an encoder which receives the input data (recognitive network) and a decoder (generative network) that generates the reconstruction of the original data. The loss of the model is the discrepancy between the original data and the reconstruction.

Specifically, it aims to find encoder ($f : \vec{R}^n \rightarrow \vec{R}^m$) and decoder functions ($g : \vec{R}^m \rightarrow \vec{R}^n$) such that the error defined as $(g(f(x))$ - $x)$ is minimized as following :

$$argmin_{f,g} E(\Delta g(f(x).\ x))$$

The E is the expectation of the distribution and Δ is the difference function, where $l^2 norm$ is commonly used. A popular form of autoencoder is given below :

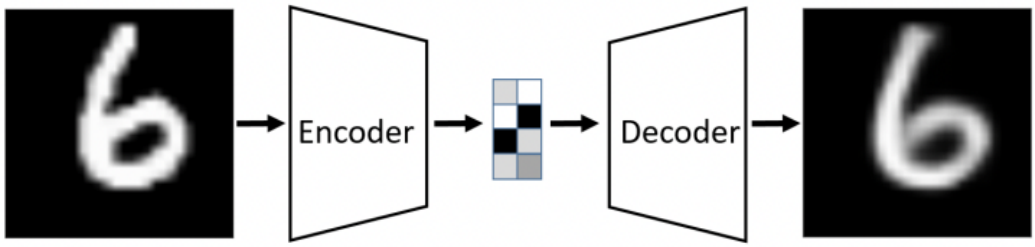


Figure 1.AutoEncoder's Encoder Decoder structure[6]

Autoencoder is used in various cases, such as image compression, dimensionality reduction, denoising of image, anomaly detection, and feature extraction.

[5] Dor Bank, Noam Koenigstein, Raja Giryes, *Autoencoders*, arXiv:2003.05991v2 [cs.LG] 3 Apr 2021
[6] Dor Bank, Noam Koenigstein, Raja Giryes, *Autoencoders*, arXiv:2003.05991v2 [cs.LG] 3 Apr 2021

4.4 Semantic Segmentation

Semantic segmentation is classifying objects in images into the same class of objects. It is also called pixel-level classification because it segments each pixel into a certain group based on the objects the pixel represents. There have been numerous approaches in semantic segmentation, including ones using K-means or SVM, and recent advancement in convolutional neural network has been combined with semantic segmentation to create remarkable improvement.[7]

This paper uses the transfer learning method to build a semantic segmentation model. Transfer learning is a method of reusing a pre-trained model, its weight, or the architecture on a new problem. This often has the advantages of reduction in computational resources and high performance because the pre-trained model is usually trained with great computational power on an extensive dataset. Unet is applied in our paper, among many semantic segmentation architectures.

The input images and their corresponding segmentation maps are used to train Unet using a convolutional neural network with stochastic gradient descent. The loss to be minimized in this model is based on the soft-max of the classification of each pixel. The soft-max is defined as

$$P_k(x) = exp(a_k(x)) / \sum_{k=1}^{K} exp(a_k(x))$$

where $a_k(x)$ equals to activation of a pixel x in class k when the image is segmented into K categories. This soft max value is aggregated using the cross-entropy as

$$E = \sum_{x \in \Omega} w(x) \log P_{l(x)}(x)$$

where $w(x)$ is the weight map to assign more importance to specific pixels and $l(x)$ is the true label of each pixel.[8] Below is an illustration of general semantic segmentation that classified each pixel according to the object type such as human, car, tree etc.

[7] Xiaolong Liu, Zhidong Deng, Yuhan Yang, *Recent progress in semantic image segmentation,* Artificial Intelligence Review

[8] Olaf Ronneberger, Philipp Fischer, Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation* , arXiv:1505.04597v1 [cs.CV] 18 May 2015

 [9]

Photo5. Example result of semantic segmentation

5. Methodology

5.1 Preprocessing

The original images without head annotation were feature data and the heatmap images were the target data. The given images are original images of crowds and the images with markers on the head. Marked images were preprocessed into a heatmap using a Gaussian kernel. Sigma was fixed on a single integer. The outcome is a heatmap image where densely crowded parts are highlighted. The heatmap images were then saved under the same name as the correponding original image into a new folder called "heatmap", in png format. Such systematic naming of files was an essential step for successful modeling by matching the right input data and the target data. Both images were resized into a square of the same size. In the training and testing of the model, a function was created and called to match the related files.

Photo6 . Original images with head annotation

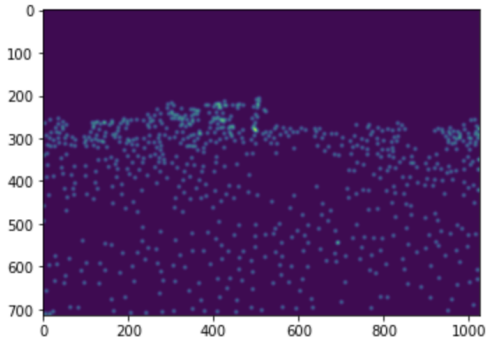


Photo7. Gaussian filtered annotation (heatmap)

5.2 Experiment 1. Auto Encoder

[9] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, *ICNet for Real-Time Semantic Segmentation on High-Resolution Images*

5.2.1 Model structure

Convolutional AutoEncoder was deployed in order to make a deep learning model that returns a reconstructed version of the original image into a heatmap, where the densely populated parts are highlighted. This model aims to solve a regression problem where the pixel value of the heatmap is the target. The model is trained to minimize the difference between the generated heatmap and the given heatmap. Original images were used as the training image and the corresponding heatmap were used as the target image. The loss of the Auto Encoder was the difference between the original image and the heatmap, thus the difference of the converted image and the heatmap is trained to be minimized. The difference or loss is calculated using the l-2 norm. The model is made of the sequential model with encoder and decoder, each part made of 12 layers. A total of 346,241 parameters were trained.

5.2.2 Training Process

Total epochs were 100 times and each epoch took around 130 seconds. Loss is the mean squared error, or $l_2 norm$ that has been discussed in part 4.2. There was no sign of overfitting since the training loss and the validation loss both decreased gradually.

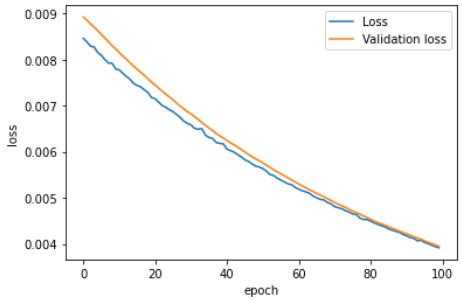


Figure2. AutoEncoder training/validation loss

5.3 Experiment 2. Semantic Segmentation

5.2.1 Model structure

As in AutoEncoder, Unet has an input and output layer that generates the output image. This model solves a classification problem since it assigns each pixel with a certain class of objects to cluster the pixel into. Thus, the error, or loss, in this model is based on a cross-entropy loss which is generally used for classification problems (detail is described in the concept section). The training data is the original picture and the target data is a heatmap as in the AutoEncoder. However, the heatmap in semantic segmentation has only two value, whether the pixel belong to a human or

not. The model is trained to produce a heatmap that is made of binary value that signifies whether a pixel belongs to the class "human". A Dynamic Unet is deployed in this project and it is made up of 14 layers.

5.2.2 Training Process

Along the epochs training loss decreased from 0.26 to 0.16. Validation loss also decreased until epoch =16 however, after that validation loss started to increase showing some degree of overfitting.

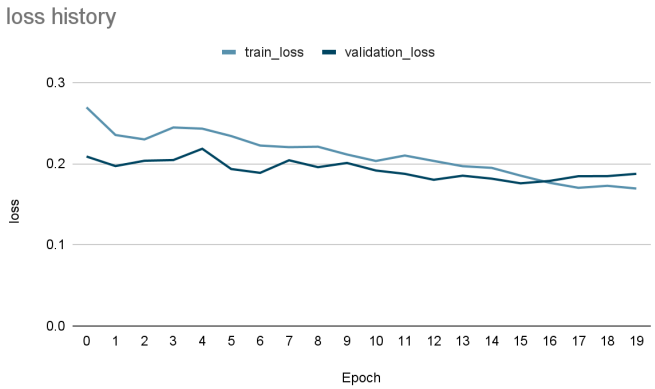


Figure3. Semantic segmentation train/validation loss

6. EVALUATION AND FINAL RESULT

6.1 Experiment 1. Auto Encoder

Since AutoEncoder solves a regression problem, mean absolute error and mean squared error was selected as evaluation criteria. The error was calculated on a test set that has not been used for training or validation of the model. Each test set outcome image's pixel values were contrasted with the target image to calculate the metric below. The pixel value takes a min value of zero and a max value of 0.003. Considering the general pixel value which ranges from 0 to 0.003, the mean absolute error of 0.03541 is relatively large. In terms of the Mean percentage error, the error is 79% [10]

	Mae	Mse	MSPE
Auto encoder	0.03541	0.00646	0.7961 (79%)

[10] In proposal, R-squared was suggested as a metric. However, in the context of pixel-wise comparison, R-squared was inappropriate. Thus, was not utilized.

Table1. Evaluation on the AutoEncoder model

Shown below are the final results of the AutoEncoder model. The model does capture human images however it also considers other objects such as trees on the street as human.

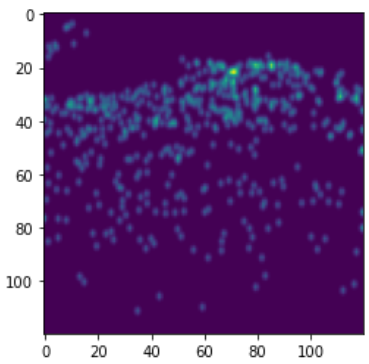
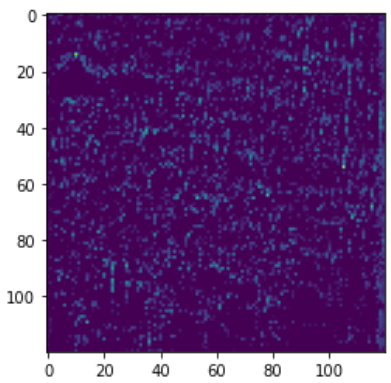
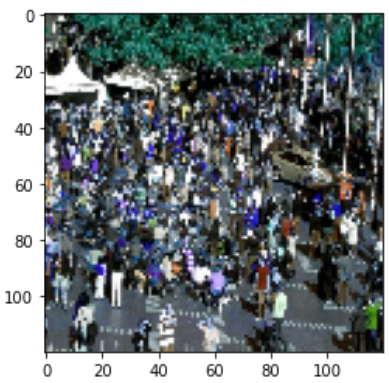
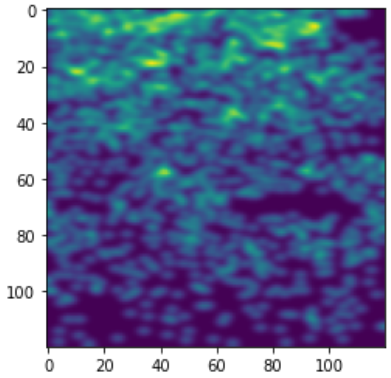
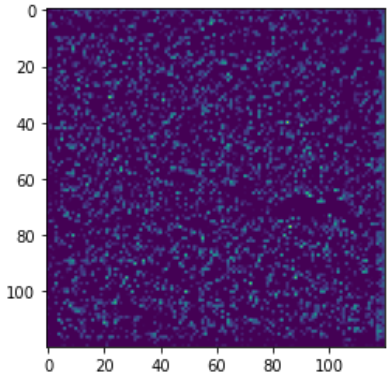
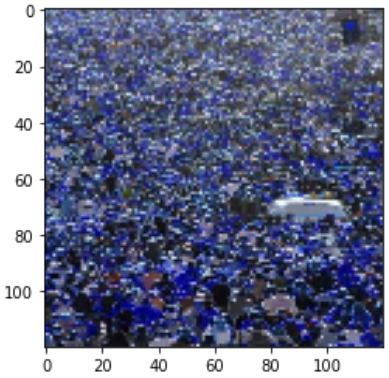
Target image	Image created by the model	Original image with marker
		
		

Figure 4. Target image / Prediction / Original image

6.2 Experiment 2. Semantic Segmentation

6.2.1 Pixel-Wise classification

Considering each pixel as one data point, this model can be evaluated as a classification problem. Hence, considering each data point in the training set and the outcome of the model, total error was calculated based on precision, recall, f1-score, and accuracy. Considering that this model intends to sensitively capture overcrowdedness, recall of class 1 would be the most significant

metric. That is because it does not increase the risk of disaster to overly detect humans, but inactive alerts can deter prevention.

This is a problem of imbalanced data where class 0, which is background is dominant (48 times more frequent than class 1). Despite that, the recall in class 1 is 0.71 meaning that around 71% of humans in the image have been properly detected and classified by the model.

Class	Precision	Recall	f1-score	Support
0 (Background)	0.99	0.97	0.98	156798
1 (Human)	0.32	0.71	0.44	3202
Accuracy	0.96			
Macro avg	0.66	0.84	0.71	160000
Weighted avg	0.98	0.96	0.97	160000

Table2. Evaluation on semantic segmentation model

Figure 5. Target image / Prediction of semantic segmentation

6.2.2 Image-wise density estimation

Regarding each image as a data point, this problem can be formulated to estimate the density of the population in images. The population was calculated as the percentage of pixels with humans within an image. In the test set, the R-squared performance of density prediction was 0.5024 and the scatter plot is as below. It shows a clear correlation between the true population density and the predicted population density by the model.

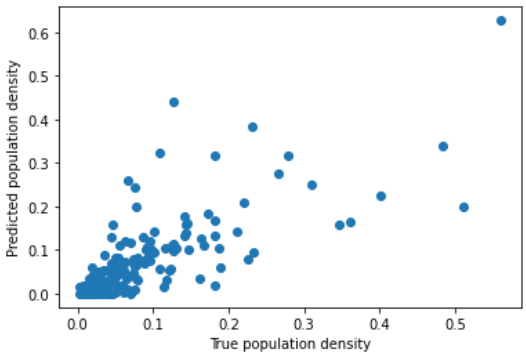


Figure 6. Density estimation result

6.2.3 Test on crawled data

As suggested in the proposal, crawled data in the real surveillance camera were tested with the developed model(Semantic segmentation model). Although we cannot exactly calculated the error since we do not have annotated image, we can see that the model is capturing the part in the image where humans are relatively densely gathered together. However, it does not find people when they are not clustered together.

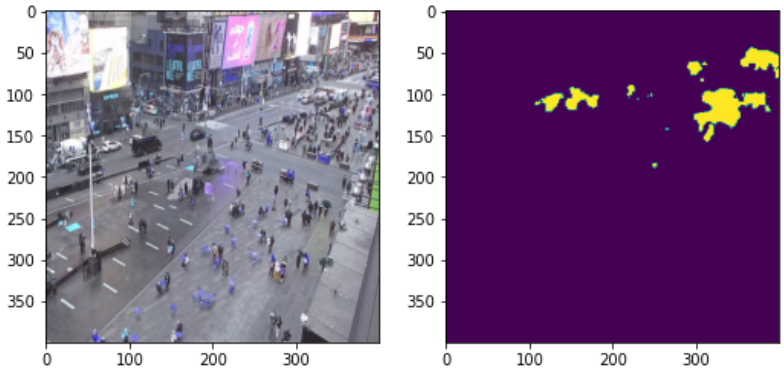


Photo 8. (Left) Original images crawled online /
(Right) Prediction of crowd density using semantic segmentation

7. CONCLUSTION, LIMITATION & FUTURE WORK

The autoencoder model was relatively unstable showing large error terms (71% mean absolute percentage error) The semantic model, which relatively showed greater performance (0.71 micro average f1-score). However, it has limitations. (1) The model performance depend on the size of people in the images. (2) Plus, the model misunderstands other round objects that resembles human heads when they are placed densely.

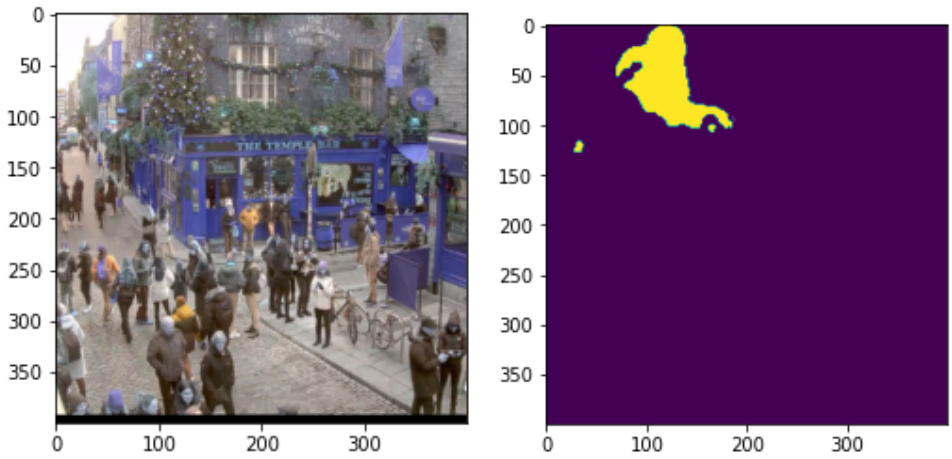


Photo9. Image of Temple bar, Dublin and its model prediction of crowd

In this image, the model did not capture heads because the size of people was larger than in training images. It rather captured balls in the tree as human heads. As the model performance depends on the size of objects, more images with various conditions should be collected and used for training the model.

8. OTHER CONSIDERATIONS: Privacy and model application

While numerous surveillance camera APIs are available publicly online in South Korea, most of them are videos of transportations for purpose of checking traffic information. Capturing the real-time image of a crowd involves privacy issues since a person's face is exposed. Hence, a public API of surveillance cameras with pedestrians was not accessible in Korea. For such a model to be fully utilized, it could be launched in each Police office or National Police Agency server, which has legal rights to access such information.

9. REFERENCES

[1] Sheng Zheng, Xiaolong Li,Xiongjie Qin, *A new image denoising method based on Gaussian filter*

[2] Dor Bank, Noam Koenigstein, Raja Giryes, *Autoencoders*, arXiv:2003.05991v2 [cs.LG] 3 Apr 2021

[3] Xiaolong Liu, Zhidong Deng, Yuhan Yang, *Recent progress in semantic image segmentation*, Artificial Intelligence Review

[4] Olaf Ronneberger, Philipp Fischer, Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation* , arXiv:1505.04597v1 [cs.CV] 18 May 2015

[5] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, Jiaya Jia, *ICNet for Real-Time Semantic Segmentation on High-Resolution Images*

[6] Wold Cam, https://worldcams.tv/

[7] Keiron O'Shea, Ryan Nash, *An Introdiction to Convolutional Neural Networks, arXiv:1511.08458v2 2 Dec 2015*

[8] Techopedia, https://www.techopedia.com/definition/17833/backpropagation

[9] **Addis Abebe Assefa, Wenhong Tian, et al,** Crowd Density Estimation in Spatial and Temporal Distortion Environment Using Parallel Multi-Size Receptive Fields and Stack Ensemble Meta-Learning, *Symmetry* **2022**, *14*, 2159. https://doi.org/10.3390/sym14102159

[9] https://www.kaggle.com/code/anushkasharmaa/crowd-detection-prediction

[10] https://www.kaggle.com/code/nexusbot/shanghai-crowd-fastai-segmentation