CHAPTER 3
RESEARCH METHODOLOGY

3.1. Literature Study

Before the researcher decides to implement the fixed architecture that will be used, the researcher decided to read several scientific pieces of literature about object detection. From this step, the researcher expects roughly what architecture will be used.

3.2. Data Collection

The dataset was obtained from the Kaggle website (https://www.kaggle.com/andrewmvd/face-mask-detection). The existing dataset is used as training data for the artificial intelligence model which consists of three classes, namely wear a mask, not wear a mask & wear a mask but wrong. The total dataset is 853 images & 853 XML annotated files.

The examples of the dataset which already labeled :

- Wear a mask

Figure 3.1: Example of “wear a mask” data

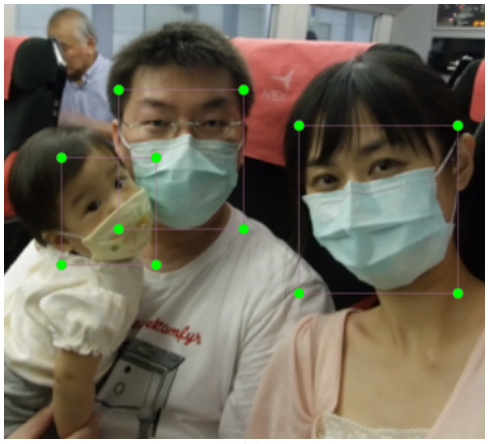

Figure 3.2: Example of “wear a mask” data

- Not wear a mask

Figure 3.3: Example of “not wear a mask” data

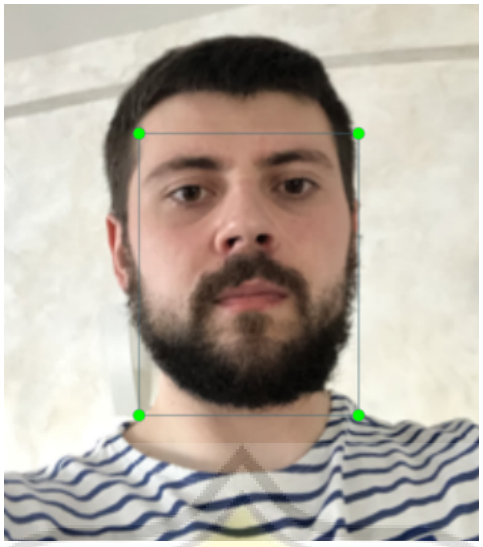

Figure 3.4: Example of “not wear a mask” data

- Wear a mask but wrong

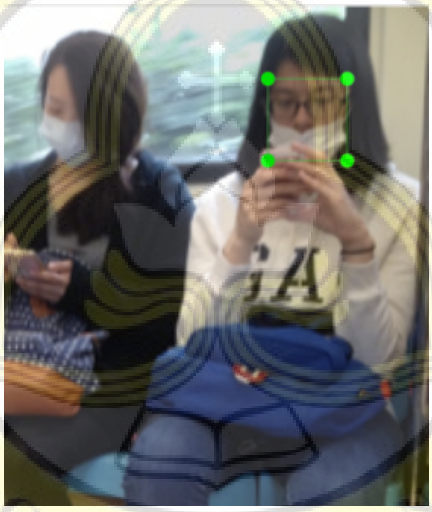

Figure 3.5: Example of “wear a mask but wrong” data

Figure 3.6: Example of “wear a mask but wrong” data

3.3. Models Config Preparation

Before the training and evaluation, the researcher needs to prepare the config that will be implemented in the framework. To all config, the researcher set 8 batch sizes for training & 5000 to the number of steps.

3.4. Conversion of the RAW Dataset to TFRecord Data

The data that was used was converted from the images to the TF Record format. Based on the TensorFlow website [17], “The TFRecord format is a basic binary record storage format”. This action made the data easier to process in the next step.

3.5. Framework Design for Training & Evaluating

The detection system is a system that is trained to detect the use of masks (Wear a mask, not wear a mask, wear a mask but wrong & back face). The researcher made one framework Python code and implement three models which consist of “Faster R-CNN ResNet50 V1 640x640” / “SSD ResNet50 V1 FPN 640x640 (RetinaNet50)” / “SSD MobileNet V2 320x320”.

1. Faster R-CNN ResNet50 V1 640x640

Faster R-CNN ResNet50 V1 is the architecture that is a mixture of the Faster R-CNN model and Resnet50 V1 model. For the image input, Faster R-CNN ResNet50 V1 using

640x640 image. Based on Ren et al. research [18], To hypothesis object locations, state-of-the-art object detection networks rely on region proposal techniques. SPPnet and Fast R-CNN have lowered the detection network's running time, exposing region proposal calculation as a bottleneck. The researchers develop a Region Proposal Network (RPN) in that paper that shares full-image convolutional features with the detection network, allowing for nearly cost-free region suggestions. An RPN is a fully convolutional network that predicts object limits and scores at each position at the same time. RPNs are trained from start to finish to create high-quality region proposals, which Fast R-CNN uses for detection. RPN and Fast R-CNN may be trained to share convolutional features using a simple alternating optimization. Their detection technique on a GPU achieves state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2 percent mAP) and 2012 (70.4 percent mAP) using 300 suggestions per image for the very deep VGG-16 model, with a frame rate of 5fps (including all steps).

Based on He et al. research [19], it's harder to train deep neural networks. The researchers present a residual learning framework that makes it easy to train considerably deeper networks than previously used networks. The researchers explicitly reformulate the layers as learning residual functions with reference to the layer inputs, rather than learning unreferenced functions. The researchers show extensive empirical evidence that residual networks are easier to modify and can acquire accuracy from a much broader depth of data. On the ImageNet dataset, the researchers evaluate residual nets with a depth of up to 152 layers, which is 8 levels deeper than VGG nets but still has a lower complexity. An ensemble of these residual nets scores 3.57 percent error on the ImageNet test set. This work took first place in the ILSVRC 2015 classification task. The researchers are also looking at CIFAR-10 with 100 and 1000 layers. The depth of representations is crucial for many visual identification tasks. Only because of our extraordinarily deep representations could the researchers achieve a 28% relative improvement on the COCO object identification dataset. Deep residual nets were used in our contributions to the ILSVRC and COCO 2015 competitions, and we won first place in the ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation tasks.

2. SSD ResNet50 V1 FPN 640x640 (RetinaNet50)

SSD ResNet50 V1 FPN is the architecture that mixes the SSD model, Resnet50 V1 model, and FPN. For the image input, SSD ResNet50 V1 FPN uses 640x640 images. Based on Liu et al. research [16], SSD (Single Shot MultiBox Detector) is a deep neural network-based approach for detecting objects in pictures. SSD discretizes the output space of bounding boxes into a set of default boxes per feature map location, using variable aspect ratios and sizes. The network generates scores for the presence of each item type in each default box at prediction time and then adjusts the box to better match the object shape. In addition, to handle objects of varied sizes naturally, the network combines predictions from numerous feature maps with different resolutions. SSD is simply a comparison to methods that use object proposals since it avoids proposal formation and subsequent pixel or feature resampling steps, encapsulating all processing in a single network. SSD is simple to train and integrate into systems that require a detecting component as a result of this. SSD offers competitive accuracy to approaches that include an additional object proposal phase and is substantially faster while providing a unified framework for both training and inference, according to experimental results on the PASCAL VOC, COCO, and ILSVRC datasets. On the VOC2007 test at 59 FPS on an Nvidia Titan X, SSD achieves 74.3 percent mAP for 300300 input, and for 512 512 input, SSD achieves 76.9% mAP, outperforming a comparable state-of-the-art Faster R-CNN model. SSD offers substantially better accuracy than other single-stage algorithms, even with a smaller input image size.

Based on He et al. research [19], it's harder to train deep neural networks. The researchers present a residual learning framework that makes it easy to train considerably deeper networks than previously used networks. The researchers explicitly reformulate the layers as learning residual functions with reference to the layer inputs, rather than learning unreferenced functions. The researchers show extensive empirical evidence that residual networks are easier to modify and can acquire accuracy from a much broader depth of data. On the ImageNet dataset, the researchers evaluate residual nets with a depth of up to 152 layers, which is 8 levels deeper than VGG nets but still has a lower complexity. An ensemble of these residual nets scores 3.57 percent error on the ImageNet test set. This work took first place in the ILSVRC 2015 classification task. The researchers are also looking at CIFAR-10

with 100 and 1000 layers. The depth of representations is crucial for many visual identification tasks. The researchers have a 28 percent success rate.

Based on Lin et al. research [20], Feature pyramids are a fundamental component of object recognition algorithms that recognize things at various scales. Because pyramid representations are computationally and memory intensive, they have been avoided in modern deep learning object detectors. The researchers leverage the intrinsic multi-scale, pyramidal structure of deep convolutional networks to generate feature pyramids at a reasonable cost in that paper. A top-down architecture with lateral links is developed for creating high-level semantic feature maps at diverse sizes. This design, known as a Feature Pyramid Network (FPN), shows substantial improvement as a generic feature extractor in a variety of applications. Their technique, which uses FPN in a primitive Faster R-CNN system, achieves state-of-the-art single-model performance on the COCO detection benchmark without bells and whistles, outperforming all existing single-model submissions, including those from the COCO 2016 competition winners. Furthermore, the system developed by the researchers can run at 6 frames per second on a GPU, making it a practical and accurate multi-scale object identification solution.

3. SSD MobileNet V2 320x320

SSD MobileNet V2 is the architecture that mixes the SSD model & MobileNet V2 model. For the input, this architecture uses 320x320 images. Based on Liu et al. research [16], SSD (Single Shot MultiBox Detector) is a deep neural network-based approach for detecting objects in pictures. SSD discretizes the output space of bounding boxes into a set of default boxes per feature map location, using variable aspect ratios and sizes. The network generates scores for the presence of each item type in each default box at prediction time and then adjusts the box to better match the object shape. In addition, to handle objects of varied sizes naturally, the network combines predictions from numerous feature maps with different resolutions. SSD is simply a comparison to methods that use object proposals since it avoids proposal formation and subsequent pixel or feature resampling steps, encapsulating all processing in a single network. SSD is simple to train and integrate into systems that require a detecting component as a result of this. SSD offers competitive accuracy to approaches that include an additional object proposal phase and is substantially faster while providing a

unified framework for both training and inference, according to experimental results on the PASCAL VOC, COCO, and ILSVRC datasets. On the VOC2007 test at 59 FPS on an Nvidia Titan X, SSD achieves 74.3 percent mAP for 300300 input, and for 512 512 input, SSD achieves 76.9% mAP, outperforming a comparable state-of-the-art Faster R-CNN model. SSD offers substantially better accuracy than other single-stage algorithms, even with a smaller input image size.

MobileNetV2 is a new mobile architecture that increases mobile models' state-of-the-art performance on a variety of workloads and benchmarks, as well as across a range of model sizes. They then show how to use a new framework called SSDLite to efficiently apply these mobile models to object identification. They also demonstrate how to utilize MobileLabv3, a reduced version of DeepLabv3, to create mobile semantic segmentation models. DeepLabv3 is based on an inverted residual structure with thin bottleneck layer shortcut connections. The intermediate expansion layer filters feature using lightweight depthwise convolutions as a source of non-linearity. We also observed that non-linearities in the thin layers must be eliminated in order to sustain representational power. They demonstrate how this improves performance and serve as the source of inspiration for this design.

3.6. Training

With the dataset and framework that have already been made, the researcher trained and recap the output of the training like computation time, memory used & classification loss. For the training, the researcher did 3 tries. For the first try, the researcher used 20% of the datasets for Training. For the second try, the researcher used 50% of the datasets for Training. For the last try, the researcher used 80% of the datasets for Training. From the output, the researcher can analyze and get some knowledge for the evaluation.

3.7. Evaluating

With the models that have already been trained, the researcher tested and recap the output of the testing like mAP, mAP Large, mAP Medium, mAP Small, AR @100, AR @100 Large, AR @100 Medium, and AR @100 Small. For the training, the researcher did 3 tries. For the first try, the researcher used 80% of the datasets for testing. For the second try, the

researcher used 50% of the datasets for testing. For the last try, the researcher used 20% of the datasets for testing.

3.8. Model Implementation to the Existing System

After testing, the most effective model was implemented to the existing system that was already developed before. The details of the existing system will be explained in the 4th chapter.