Face Mask Detection

Vu Nguyen
Shu Han
Zixuan Deng

Outline

1. **Background & Motivation**
2. **Data Used**
3. **Methodologies (CNN, Faster RCNN, YOLOv4)**
4. **Results**
5. **Conclusions**

Background & Motivation

Figure 1: An image of COVID-19 [1]

Figure 2: Mask Required Sign [2]

University of Victoria

Background & Motivation

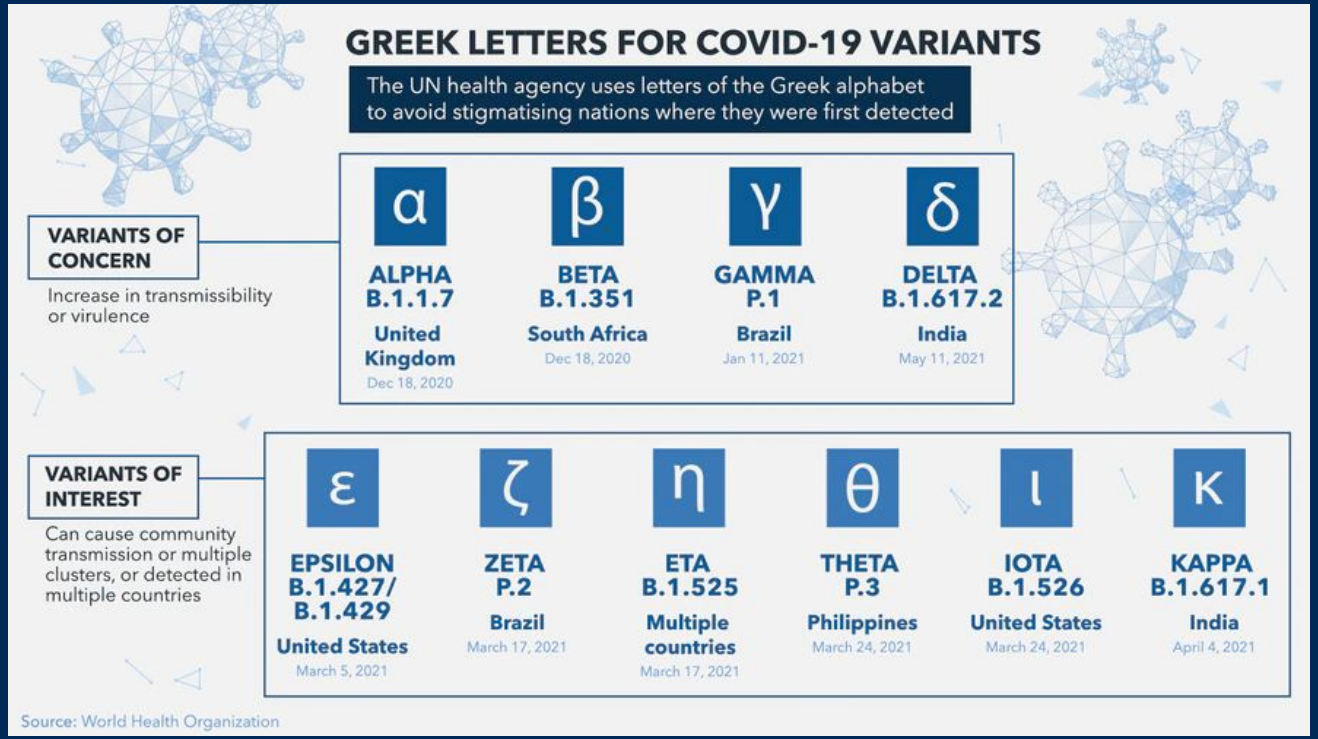


Figure 3: COVID-19 Variants [3]

University of Victoria

Data used

- CNN : A. Jangra, Face mask detection ~12k images dataset, Kaggle [4].
- Faster RCNN: Larxel, Face mask detection, Kaggle [5].
- YOLOv4: T. Zizou, Labeled mask dataset (yolo_darknet), Kaggle[6].

University of Victoria

CNN Data

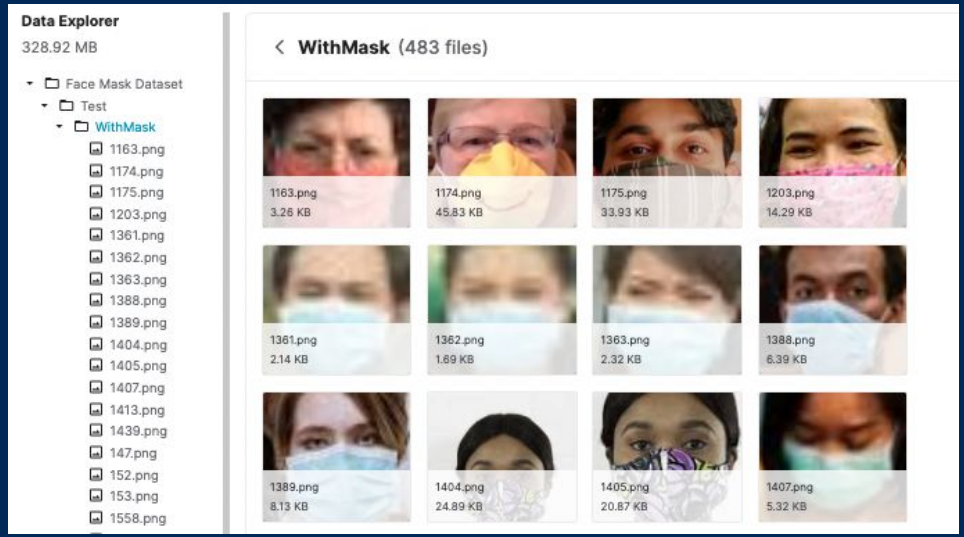


Figure 4: image sample of data [4]

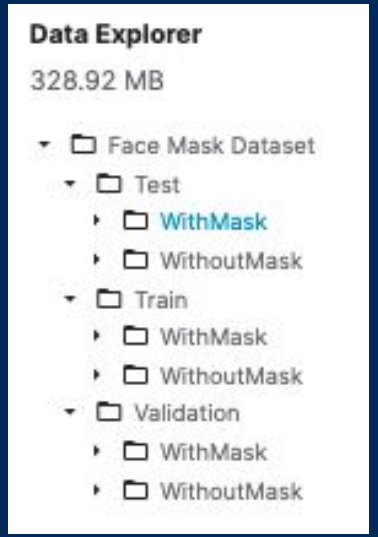


Figure 5: Structure of data [4]

University of Victoria

Faster RCNN Data

Figure 6: image sample of data [5]

Figure 7: annotations in data [5]

University of Victoria

YoloV4 Data

Figure 8: image sample of data [6]

Figure 9: annotations in data [6]

University of Victoria

Testing Data

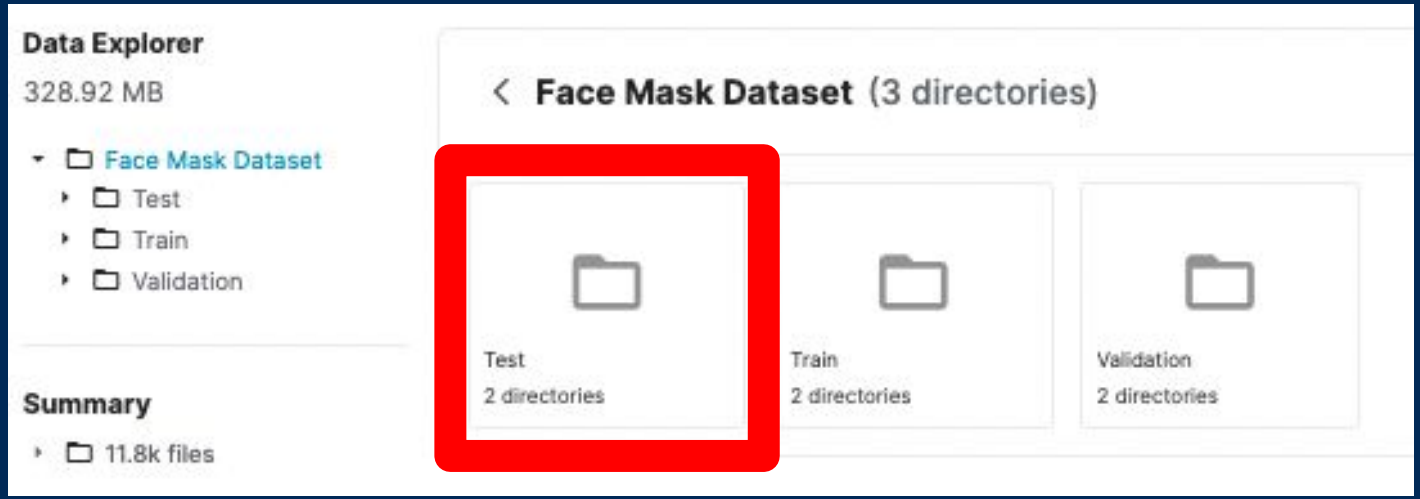


Figure 10: Testing data [4]

University of Victoria

Methodology

- CNN
- Faster RCNN
- YoloV4

University of Victoria

CNN

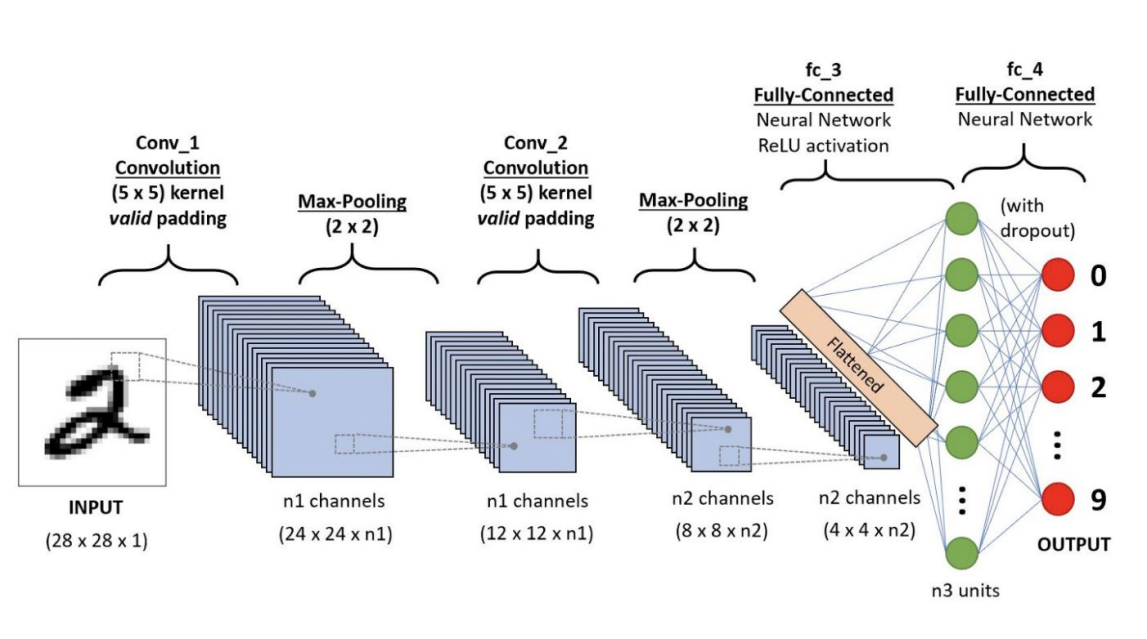


Figure 11: A CNN sequence to classify handwritten digits [7]

University of Victoria

CNN

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 126, 126, 32)      896
_________________________________________________________________
activation (Activation)      (None, 126, 126, 32)      0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 31, 31, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 31, 31, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 30752)             0
_________________________________________________________________
dense (Dense)                (None, 128)               3936384
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 3,937,409
Trainable params: 3,937,409
Non-trainable params: 0
_________________________________________________________________
```

Figure 12: CNN Model In Keras

University of Victoria

CNN Result

```
Epoch 42/50
125/125 [==============================] - 23s 186ms/step - loss: 0.0435 - a
ccuracy: 0.9939 - val_loss: 0.0793 - val_accuracy: 0.9862
Epoch 43/50
125/125 [==============================] - 23s 183ms/step - loss: 0.0947 - a
ccuracy: 0.9819 - val_loss: 0.0682 - val_accuracy: 0.9925
Epoch 44/50
125/125 [==============================] - 23s 184ms/step - loss: 0.0848 - a
ccuracy: 0.9811 - val_loss: 0.0745 - val_accuracy: 0.9887
Epoch 45/50
125/125 [==============================] - 23s 180ms/step - loss: 0.0975 - a
ccuracy: 0.9863 - val_loss: 0.1035 - val_accuracy: 0.9712
Epoch 46/50
125/125 [==============================] - 23s 187ms/step - loss: 0.0936 - a
ccuracy: 0.9800 - val_loss: 0.0693 - val_accuracy: 0.9925
Epoch 47/50
125/125 [==============================] - 23s 180ms/step - loss: 0.0669 - a
ccuracy: 0.9909 - val_loss: 0.0602 - val_accuracy: 0.9950
Epoch 48/50
125/125 [==============================] - 23s 184ms/step - loss: 0.0724 - a
ccuracy: 0.9855 - val_loss: 0.0621 - val_accuracy: 0.9925
Epoch 49/50
125/125 [==============================] - 23s 182ms/step - loss: 0.0641 - a
ccuracy: 0.9907 - val_loss: 0.0610 - val_accuracy: 0.9950
Epoch 50/50
125/125 [==============================] - 23s 185ms/step - loss: 0.0644 - a
ccuracy: 0.9908 - val_loss: 0.0609 - val_accuracy: 0.9912
```

Figure 13: CNN Training and Validation Accuracy

University of Victoria

CNN Result

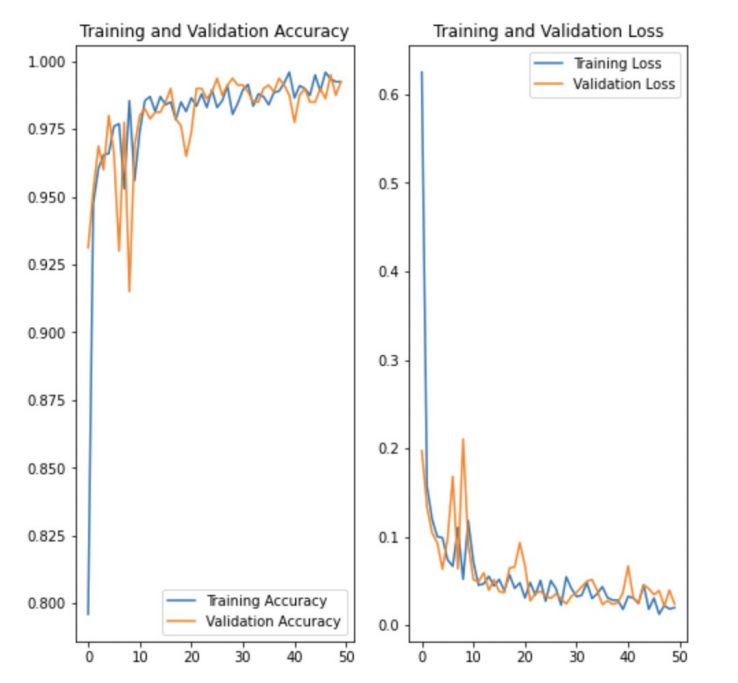


Figure 14: CNN Training and Validation Accuracy history

University of Victoria

CNN Result

	precision	recall	f1-score	support
withMask	1.00	0.97	0.99	483
withoutMask	0.98	1.00	0.99	509
accuracy			0.99	992
macro avg	0.99	0.99	0.99	992
weighted avg	0.99	0.99	0.99	992

Figure 15: CNN Result Report

University of Victoria

CNN Result

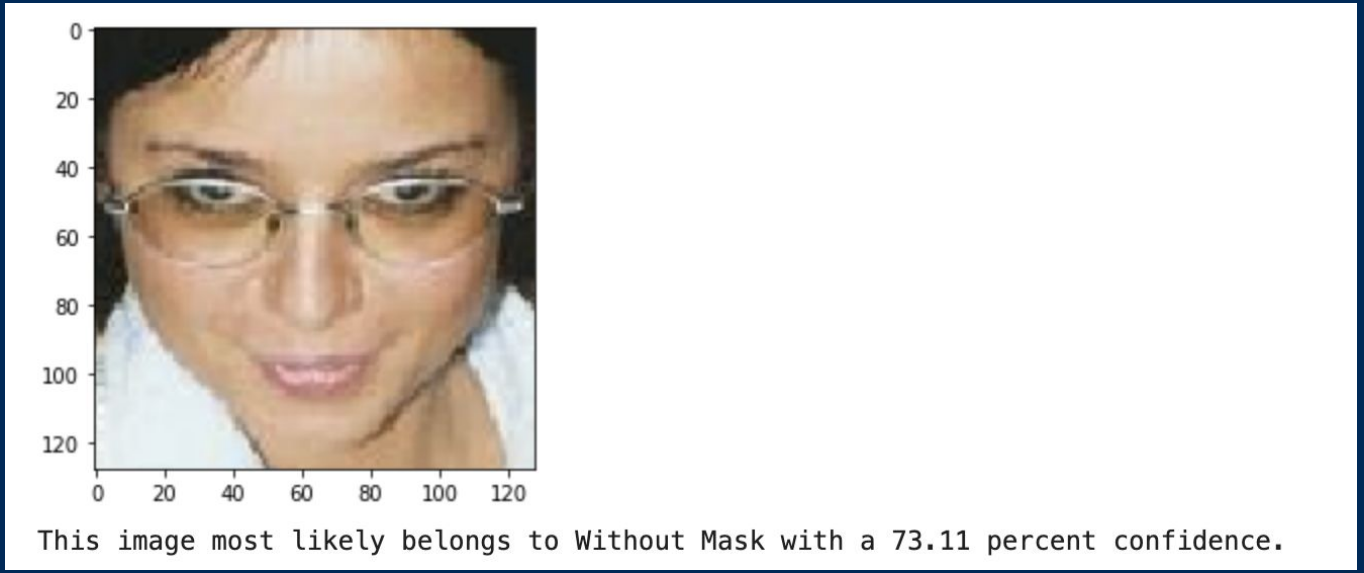
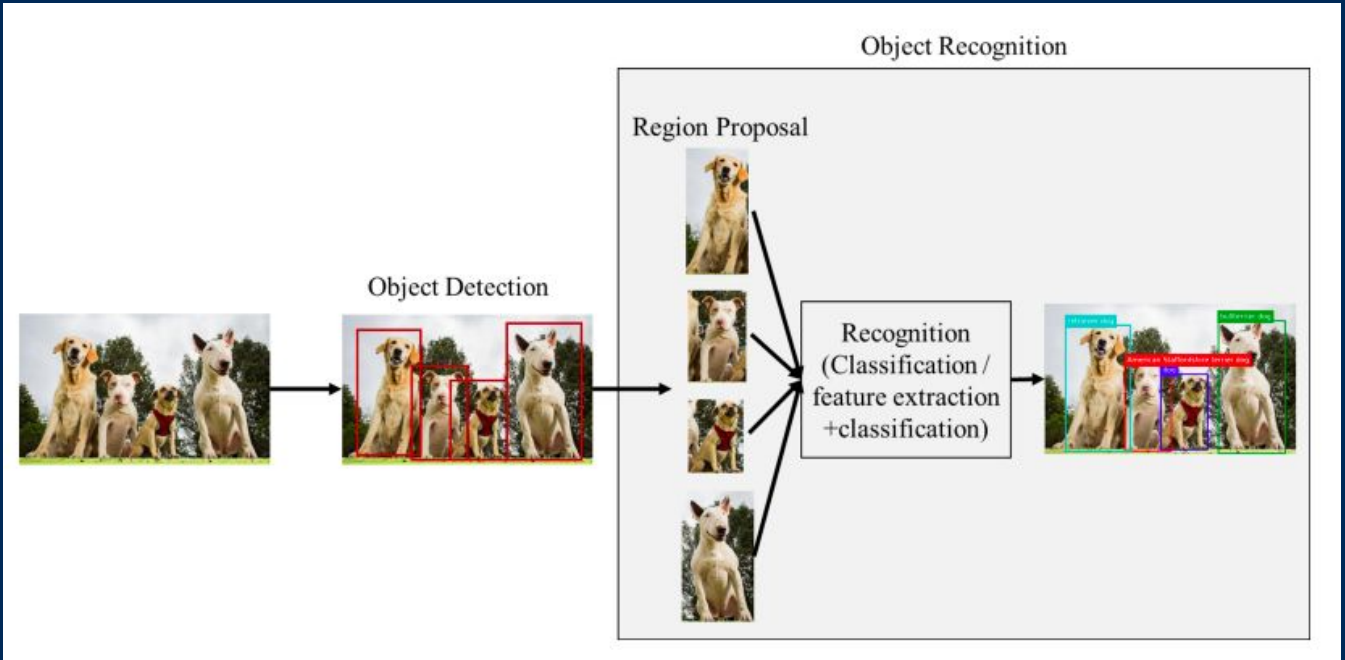


Figure 16: CNN demo prediction result

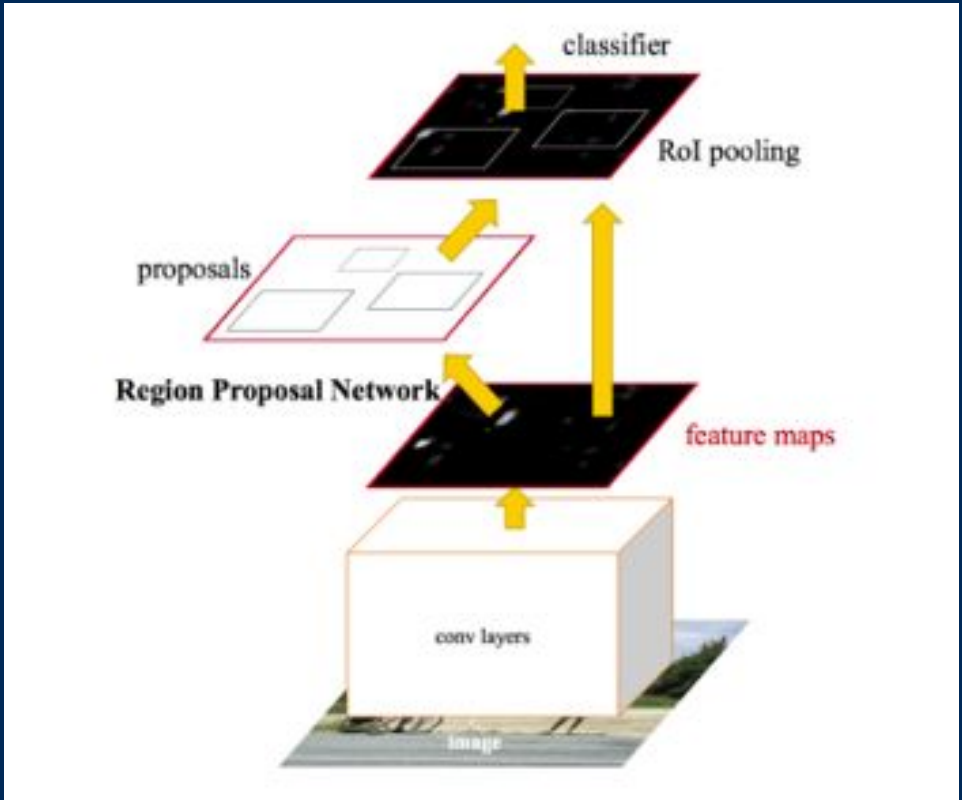
University of Victoria

Faster RCNN



University of Victoria

Region Proposal Network



University of Victoria

Faster RCNN Pytorch Model

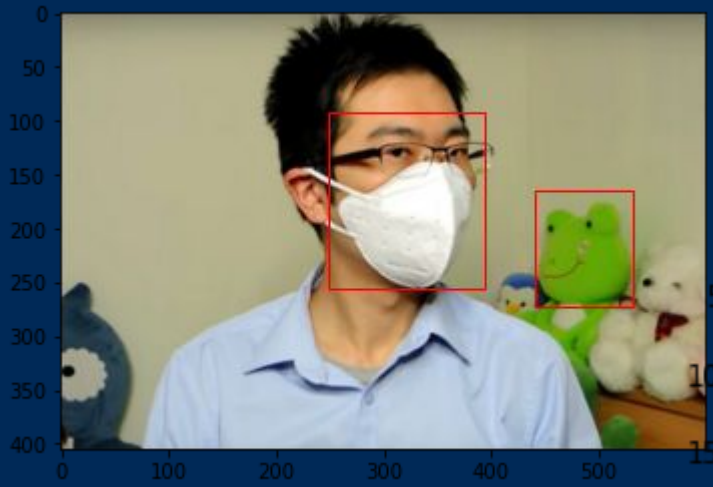
```
def get_model_instance_segmentation(num_classes):
    # load an instance segmentation model pre-trained pre-trained on COCO
    model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
    # get number of input features for the classifier
    in_features = model.roi_heads.box_predictor.cls_score.in_features
    # replace the pre-trained head with a new one
    model.roi_heads.box_predictor = FastRCNNPredictor(in_features, num_classes)

    return model
```

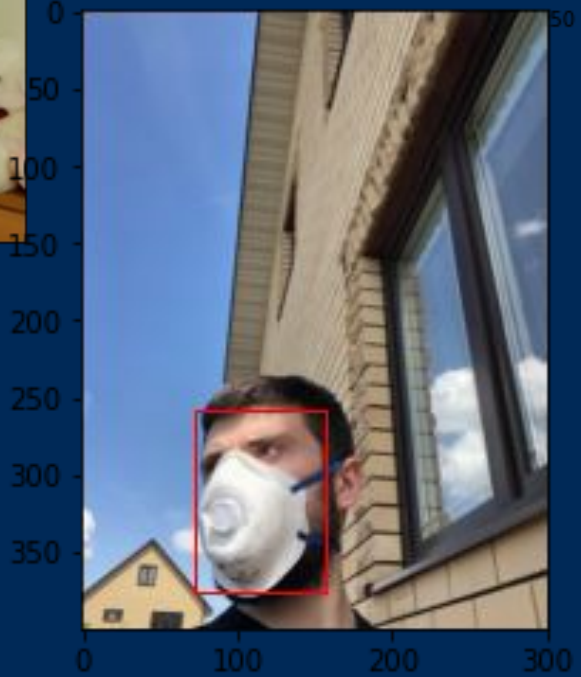
```
model = get_model_instance_segmentation(3)
```

Faster RCNN Result

```
tensor(55.8141, device='cuda:0', grad_fn=<AddBackward0>)
tensor(30.8253, device='cuda:0', grad_fn=<AddBackward0>)
tensor(22.7319, device='cuda:0', grad_fn=<AddBackward0>)
tensor(19.6140, device='cuda:0', grad_fn=<AddBackward0>)
tensor(15.5099, device='cuda:0', grad_fn=<AddBackward0>)
tensor(13.4772, device='cuda:0', grad_fn=<AddBackward0>)
tensor(13.9610, device='cuda:0', grad_fn=<AddBackward0>)
tensor(10.0978, device='cuda:0', grad_fn=<AddBackward0>)
tensor(9.3663, device='cuda:0', grad_fn=<AddBackward0>)
tensor(9.5361, device='cuda:0', grad_fn=<AddBackward0>)
```

Faster RCNN Result





University of Victoria

YOLOv4

YOLOv4 stands for You Only Look Once Version 4 which is an object detection algorithm introduced in 2020, and it is also an improved version based on its previous YOLOv3.

At present, object detection algorithms based on deep learning are usually divided into two categories.

- Two-Stage algorithm: based on the R-CNN and TridenNet, etc. Two-stage detectors decouple the task of object localization and classification for each bounding box.
- One-Stage algorithm: based on the SSD and YOLO, which has high real-time performance in multi-scale object detection. One-stage detectors make the predictions for object localization and classification at the same time.

University of Victoria

YOLOv4

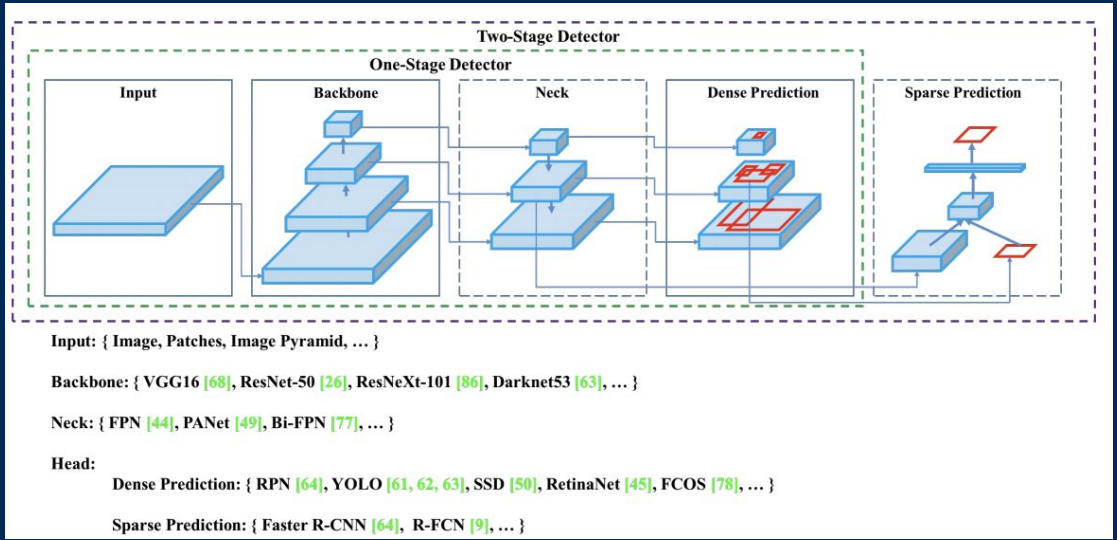


Figure 15: Object Detector [8]

The final structure of YOLOv4:

- Backbone: CSPDarknet53
- Neck: SPP , PAN
- Head: YOLOv3

Based on the speed and accuracy [8]

University of Victoria

YoloV4

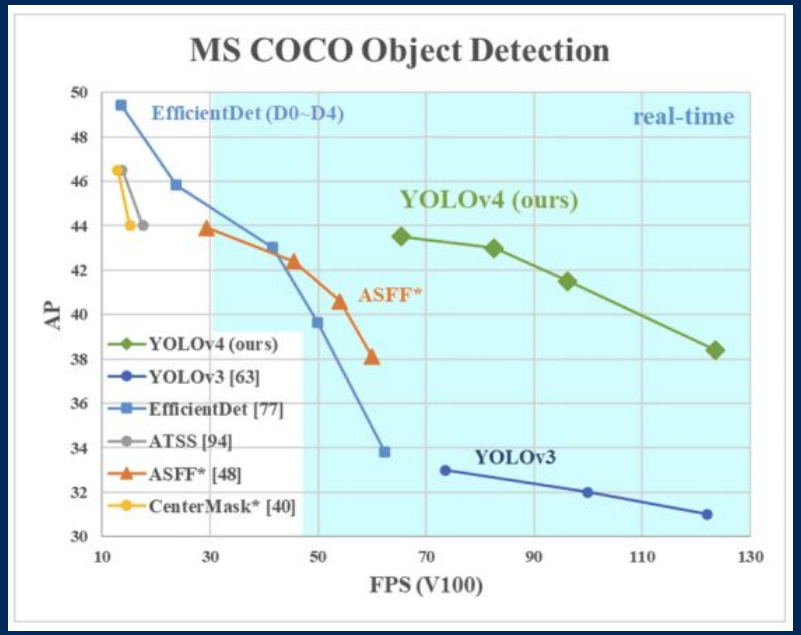


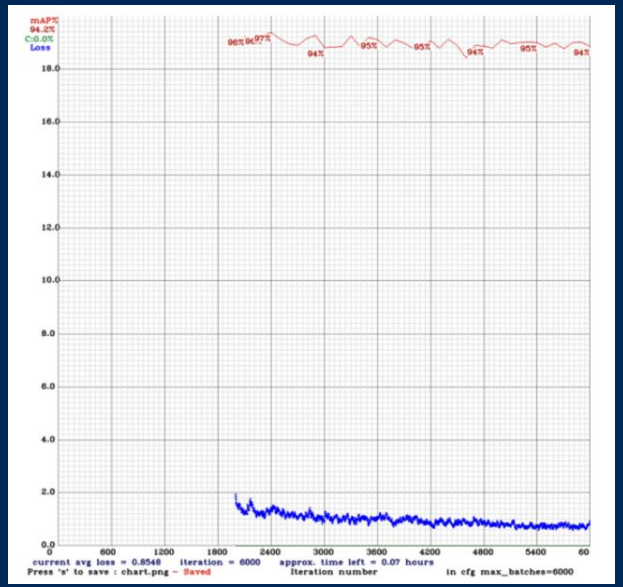
Figure 16: Algorithm Comparison [8]

University of Victoria

YOLOv4

My Drive > yolov4 > training

Name	Owner	Last modified	File size
yolov4-custom_1000.weights	me	12:58 PM me	244.2 MB
yolov4-custom_2000.weights	me	12:59 PM me	244.2 MB
yolov4-custom_3000.weights	me	1:55 PM me	244.2 MB
yolov4-custom_4000.weights	me	2:39 PM me	244.2 MB
yolov4-custom_5000.weights	me	3:22 PM me	244.2 MB
yolov4-custom_6000.weights	me	4:07 PM me	244.2 MB
yolov4-custom_best.weights	me	1:27 PM me	244.2 MB
yolov4-custom_final.weights	me	4:07 PM me	244.2 MB
yolov4-custom_last.weights	me	4:08 PM me	244.2 MB



University of Victoria

YOLOv4

```
!./darknet detector test data/obj.data cfg/yolov4-custom.cfg /mydrive/yolov4/training/yolov4-custom_best.weights /mydrive/mask_test_images/image1.jpg -thresh 0.3
imShow('predictions.jpg')
 160 conv     21       1 x 1/ 1      13 x  13 x1024 ->   13 x  13 x  21 0.007 BF
 161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, obj_norm: 1.00, cls_norm: 1.00, delta_norm: 1.00, scale_x_y: 1.05
nms_kind: greedynms (1), beta = 0.600000
Total BFLOPS 59.570
avg_outputs = 489910
 Allocate additional workspace_size = 52.43 MB
Loading weights from /mydrive/yolov4/training/yolov4-custom_best.weights...
 seen 64, trained: 153 K-images (2 Kilo-batches_64)
Done! Loaded 162 layers from weights-file
 Detection layer: 139 - type = 28
 Detection layer: 150 - type = 28
 Detection layer: 161 - type = 28
/mydrive/mask_test_images/image1.jpg: Predicted in 9.507000 milli-seconds.
with_mask: 97%
with_mask: 99%
with_mask: 98%
with_mask: 99%
with_mask: 99%
with_mask: 100%
with_mask: 99%
with_mask: 95%
with_mask: 100%
with_mask: 99%
with_mask: 98%
Unable to init server: Could not connect: Connection refused

(predictions:783882): Gtk-WARNING **: 00:22:14.306: cannot open display:
```

```
/mydrive/mask_test_images/image14.jpg: Predicted in 18.509000 milli-seconds.
with_mask: 98% (left_x:  54  top_y:   1  width:  57  height:  51)
without_mask: 100%  (left_x: 118  top_y:  39  width:  87  height: 104)
with_mask: 99% (left_x: 224  top_y:  40  width: 101  height: 109)
with_mask: 99% (left_x: 442  top_y:   9  width: 142  height: 143)
Enter Image Path:  Detection layer: 139 - type = 28
```

University of Victoria

Conclusion

Model	Training Time	Detection Time	Accuracy
CNN	Around 13 mins	Few seconds	99%
Faster RCNN	Around 1 hour	Few seconds	91%
YOLOv4	Around 5 hours	Around 15 ms	96%

University of Victoria

References:

[1] “THE CORONAVIRUS MAY BE NOVEL, BUT PREPARING FOR A LOCAL OUTBREAK ISN'T ANYTHING NEW FOR OCEAN COUNTY HEALTH DEPARTMENT,” Ocean County Health Department. [Online]. Available: https://www.ochd.org/2020/03/03/the-coronavirus-may-be-novel-but-preparing-for-a-local-outbreak-isnt-anything-new-for-ocean-county-health-department/. [Accessed: 28-Jul-2021].

[2] “Masks,” Island Health. [Online]. Available: https://www.islandhealth.ca/learn-about-health/covid-19/masks. [Accessed: 28-Jul-2021].

[3] World Health Organization, “Infection Numbers Continue to Decline, but COVID-19 Response Team Warns About Variants,” Voice of America, 08-Jun-2021. [Online]. Available: https://www.voanews.com/covid-19-pandemic/infection-numbers-continue-decline-covid-19-response-team-warns-about-variants. [Accessed: 28-Jul-2021].

[4] A. Jangra, “Face Mask Detection ~12K Images Dataset,” Kaggle, 26-May-2020. [Online]. Available: https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset. [Accessed: 28-Jul-2021].

[5] Larxel, “Face Mask Detection,” Kaggle, 22-May-2020. [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. [Accessed: 28-Jul-2021].

[6] T. Zizou, “Labeled Mask Dataset (YOLO_darknet),” Kaggle, 07-Feb-2021. [Online]. Available: https://www.kaggle.com/techzizou/labeled-mask-dataset-yolo-darknet. [Accessed: 28-Jul-2021].

[7] S. Saha, “A Comprehensive Guide to Convolutional Neural Networks-the ELI5 way,” Medium, 17-Dec-2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 28-Jul-2021].

[8] A. Bochkovskiy, C. Wang and H. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 20

University of Victoria

Questions?

University of Victoria