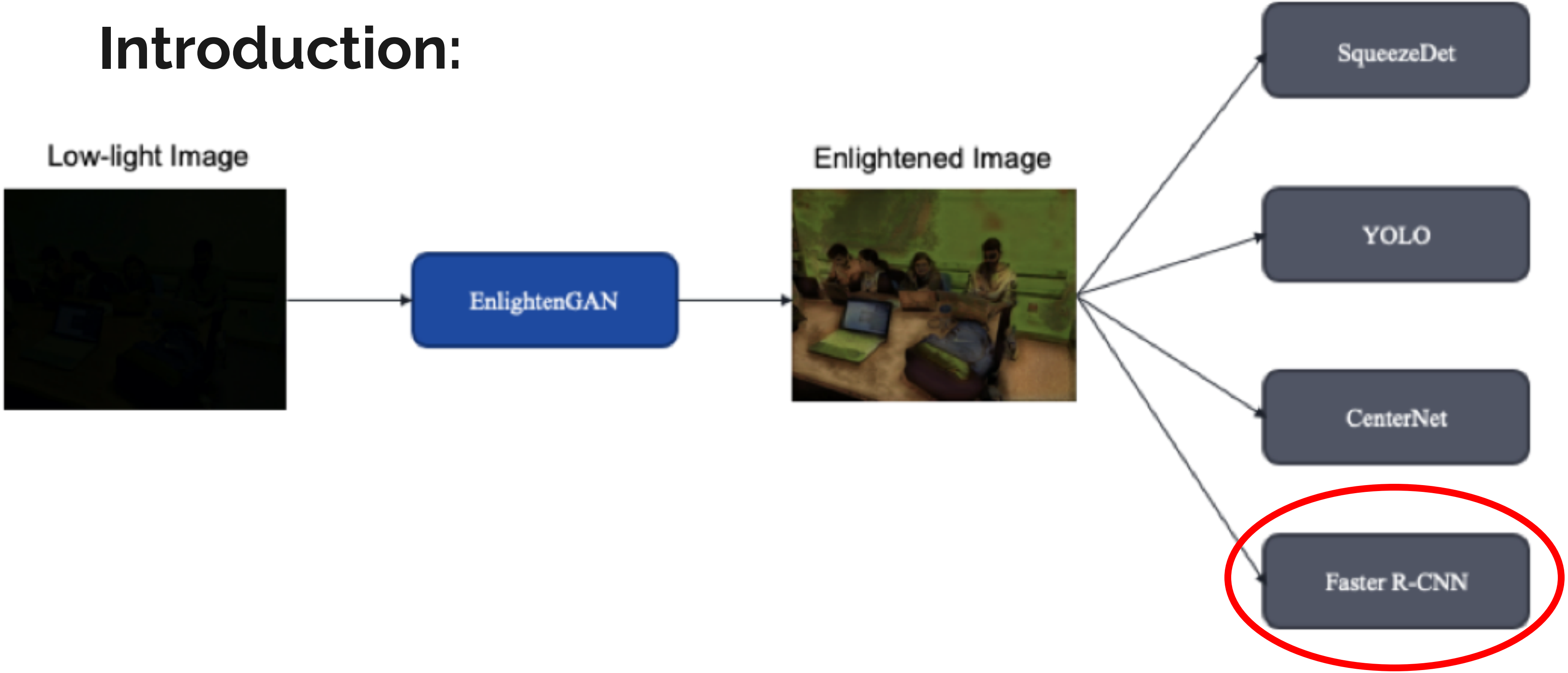
Comparative Evaluation of FineTuned Faster R-CNN Model on Dark Images Using Different Architectures

Anket Sah
Amala Chirayil
Ksheeraj Vepuri
Kriti Gupta
Sanmesh Bhosale

CS 256
12/09/19

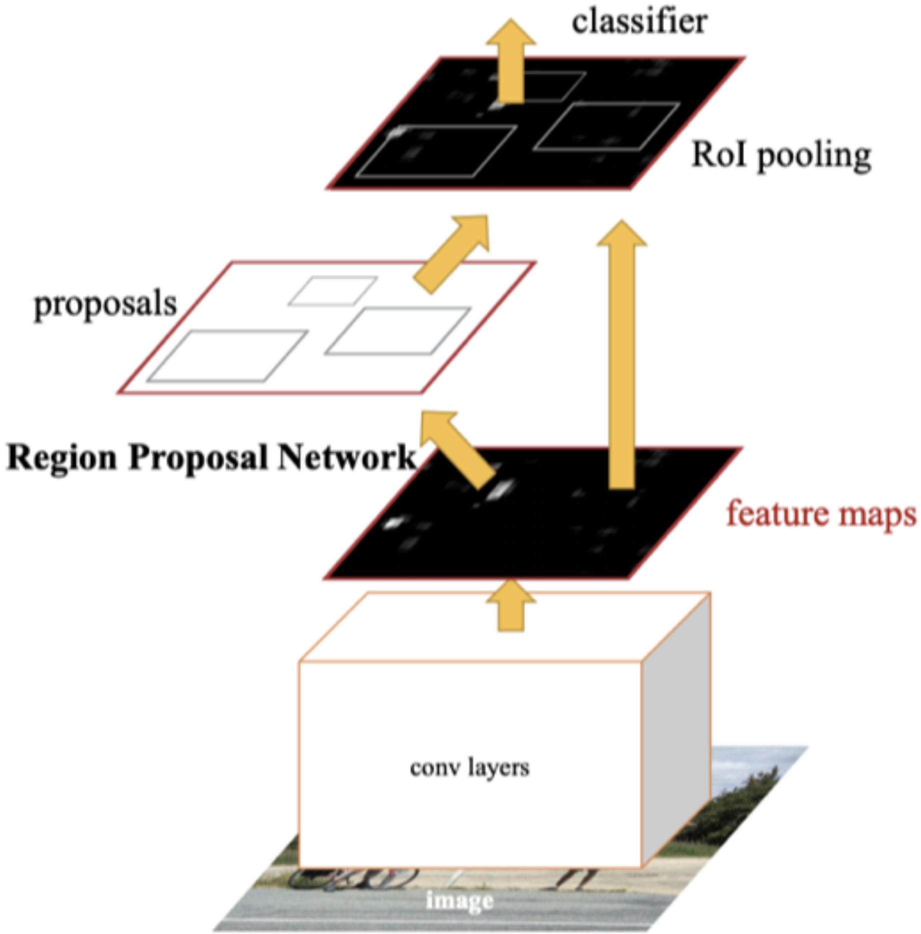
Introduction:



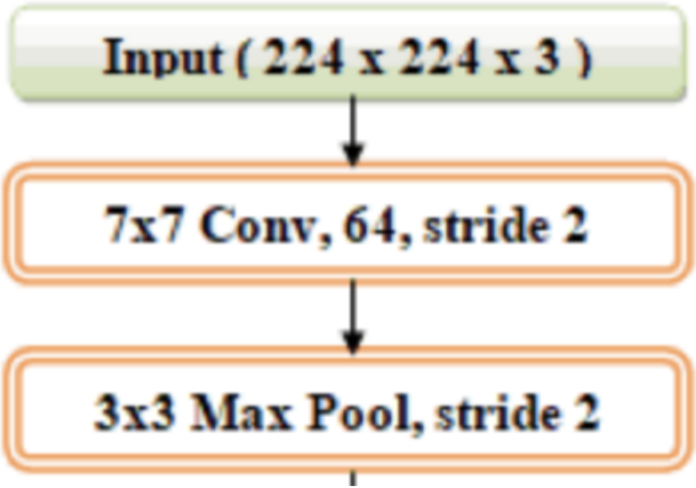
Faster R-CNN

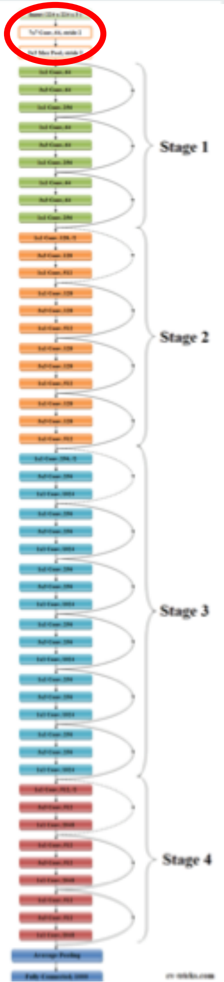
- Composed of 3 neural networks:
  - Feature Network
  - Region Proposal Network (RPN)
  - Detection Network

- Specifications (Detectron2):
  - Resnet50
  - Feature Pyramid Network (FPN)

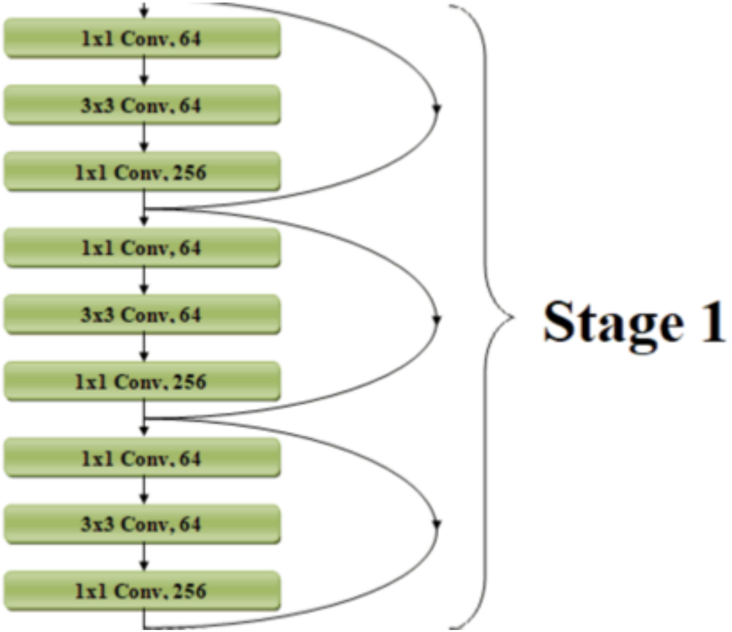


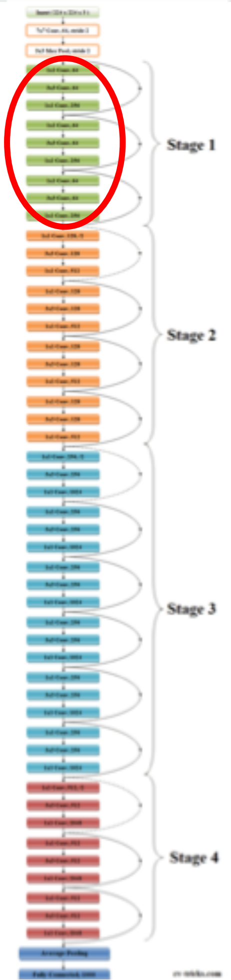
Residual Network- 50 (ResNet50)



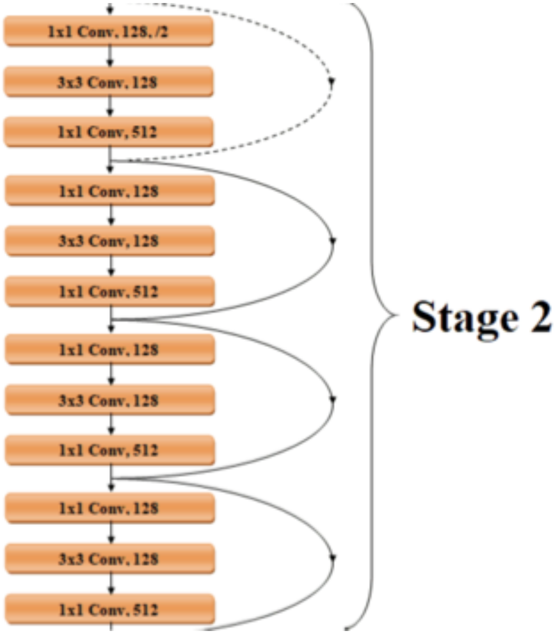


Residual Network- 50 (ResNet50)

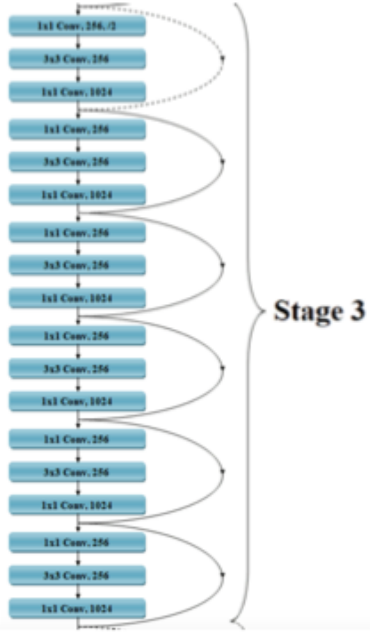




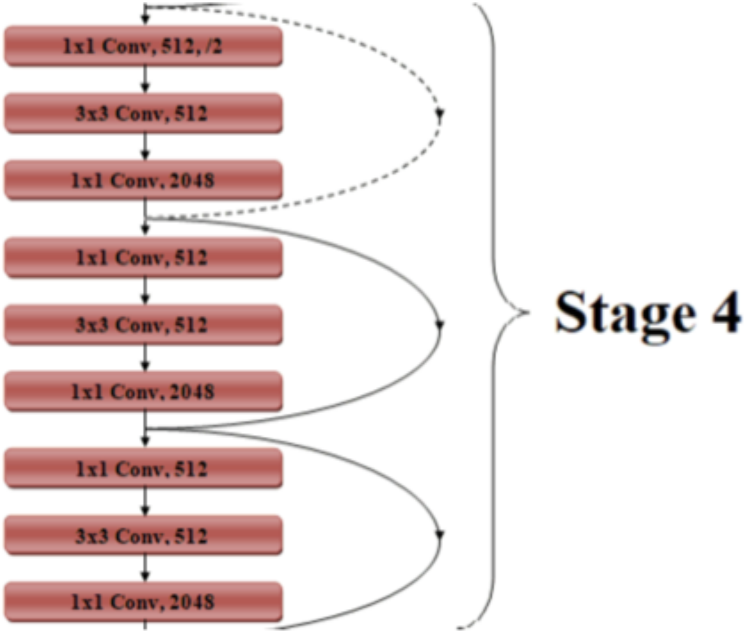
Residual Network- 50 (ResNet50)

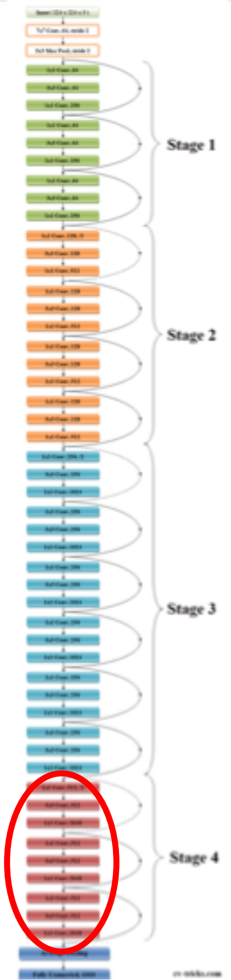


Residual Network- 50 (ResNet50)

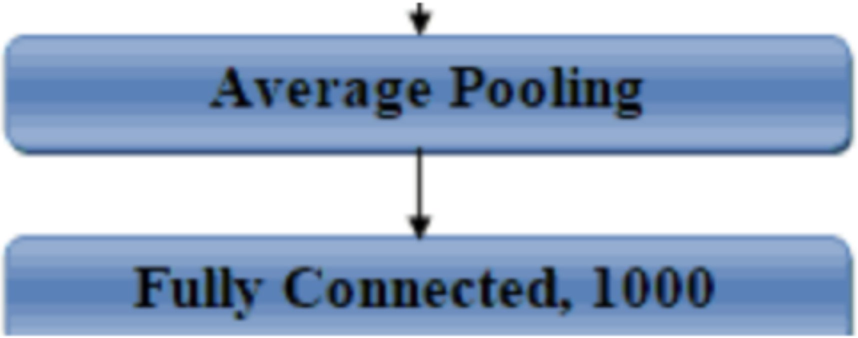


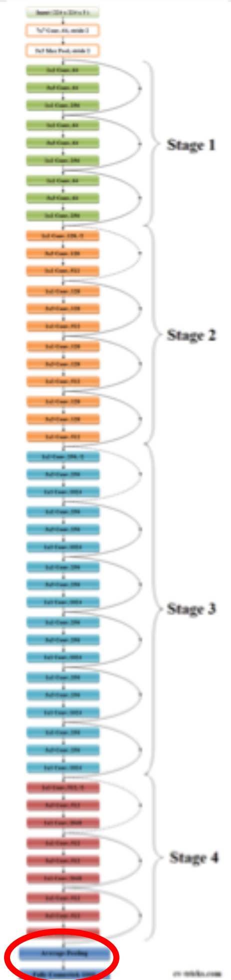
Residual Network- 50 (ResNet50)



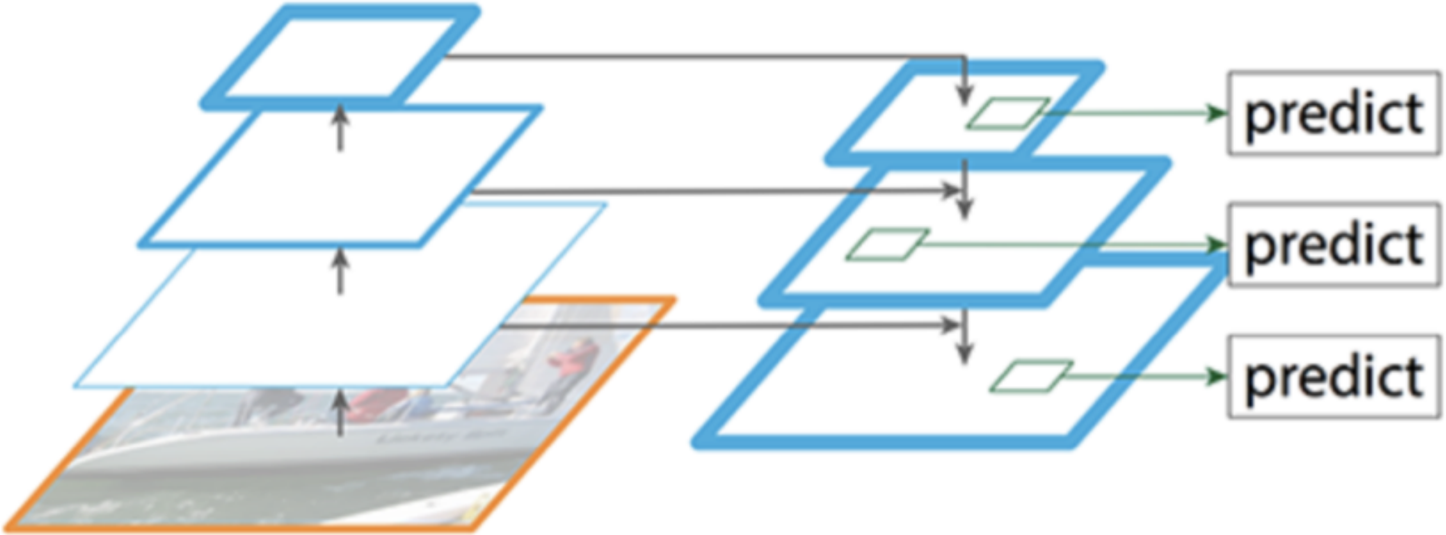


Residual Network- 50 (ResNet50)

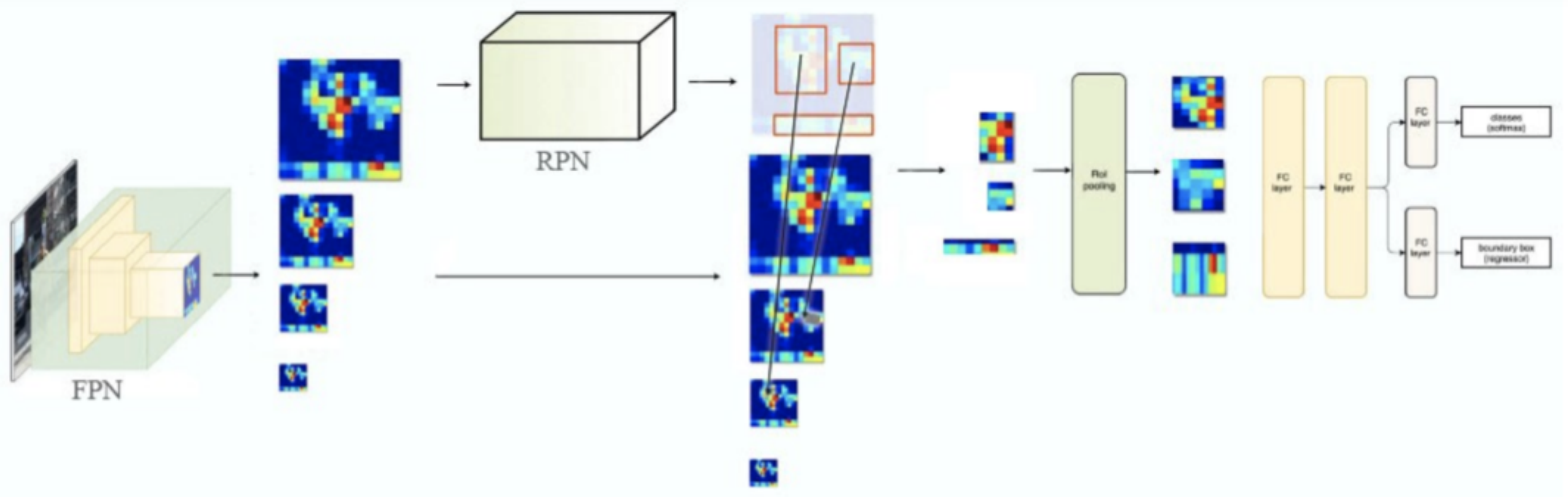




Feature Pyramid Network (FPN):



Faster R-CNN Model:

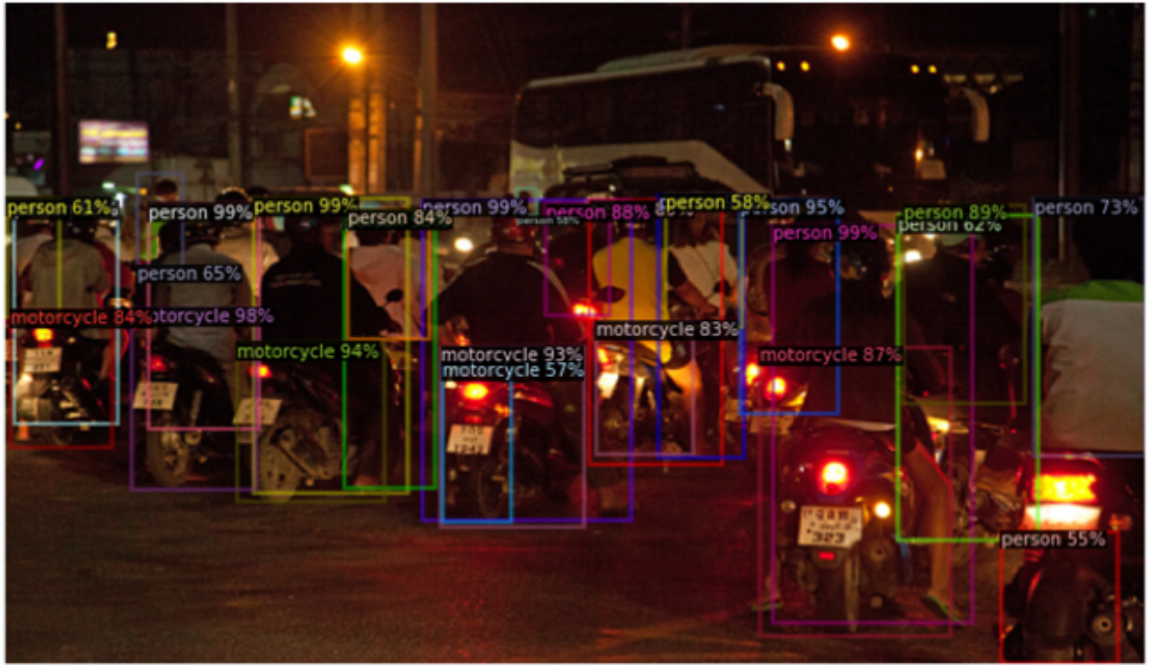


Implementation Details

- AWS EC2 Instance with Deep Learning Base AMI (Ubuntu 16.04) and version 20.0 with p2.xlarge GPU instance was setup in order to run this model.
- The implementation code that we incorporated used Detectron2.
- p2.xlarge specifications
  - GPU -1
  - vCPUs - 4
  - RAM - 61GB
- Software specifications
  - Python 3.7
  - CUDA 10.0
  - cnDNN 7.6.4
  - PyTorch 1.3.0
  - TensorFlow 1.5.0rc2
  - Keras 2.2.5
  - MxNet 1.6.0
- Code Reference: https://github.com/facebookresearch/detectron2

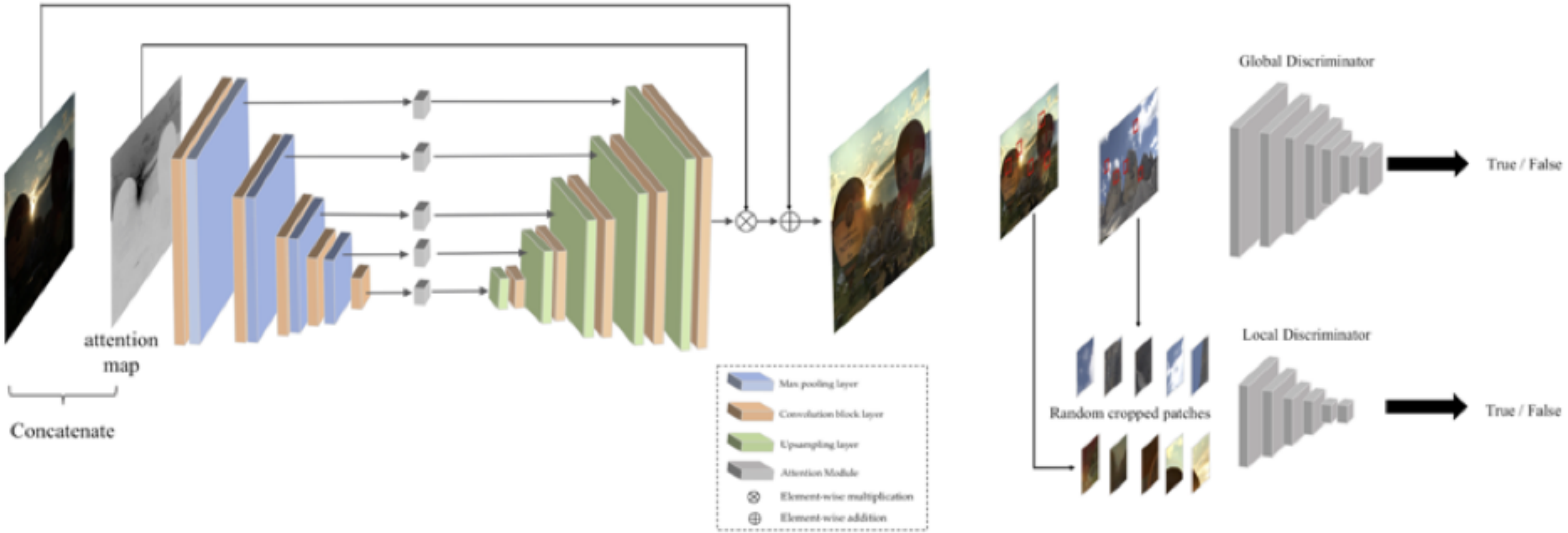
Architecture Diagram 1:

Architecture Diagram 1: Output

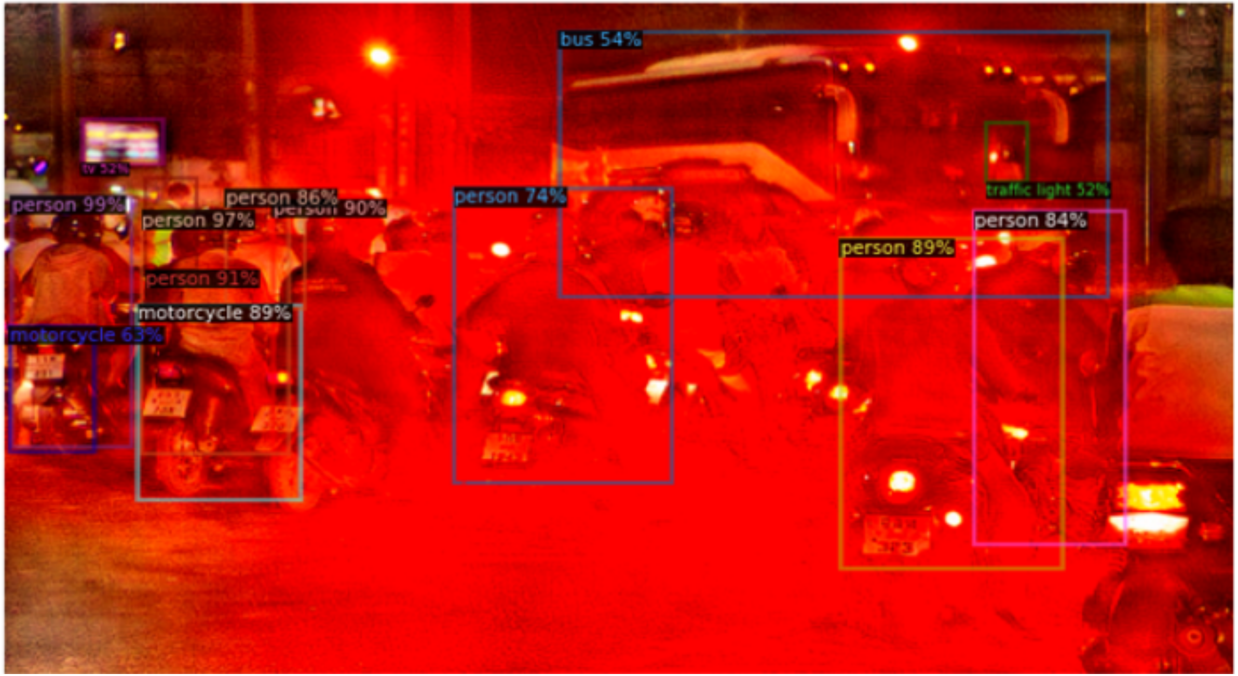


Architecture Diagram 2:

EnlightenGAN (EG)



Architecture Diagram 2: Output

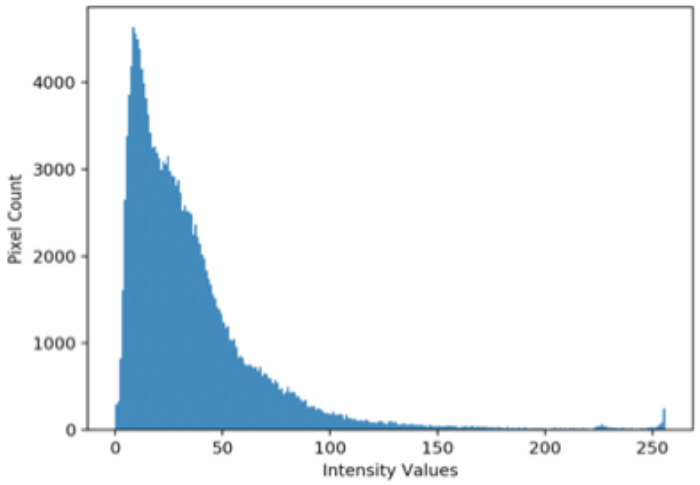


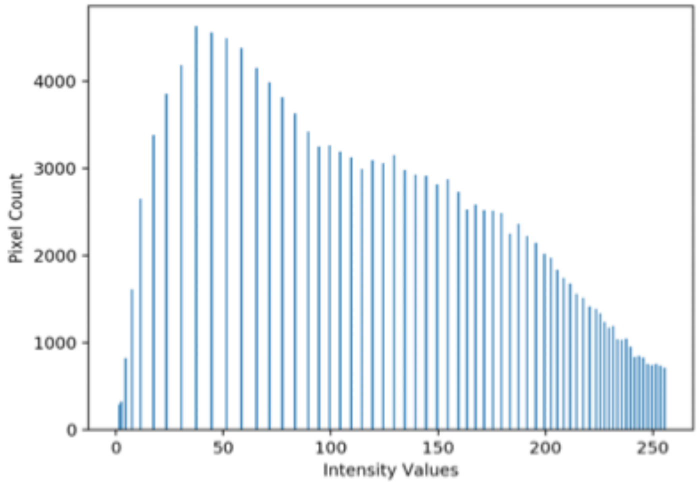
Architecture Diagram 3:

CLAHE

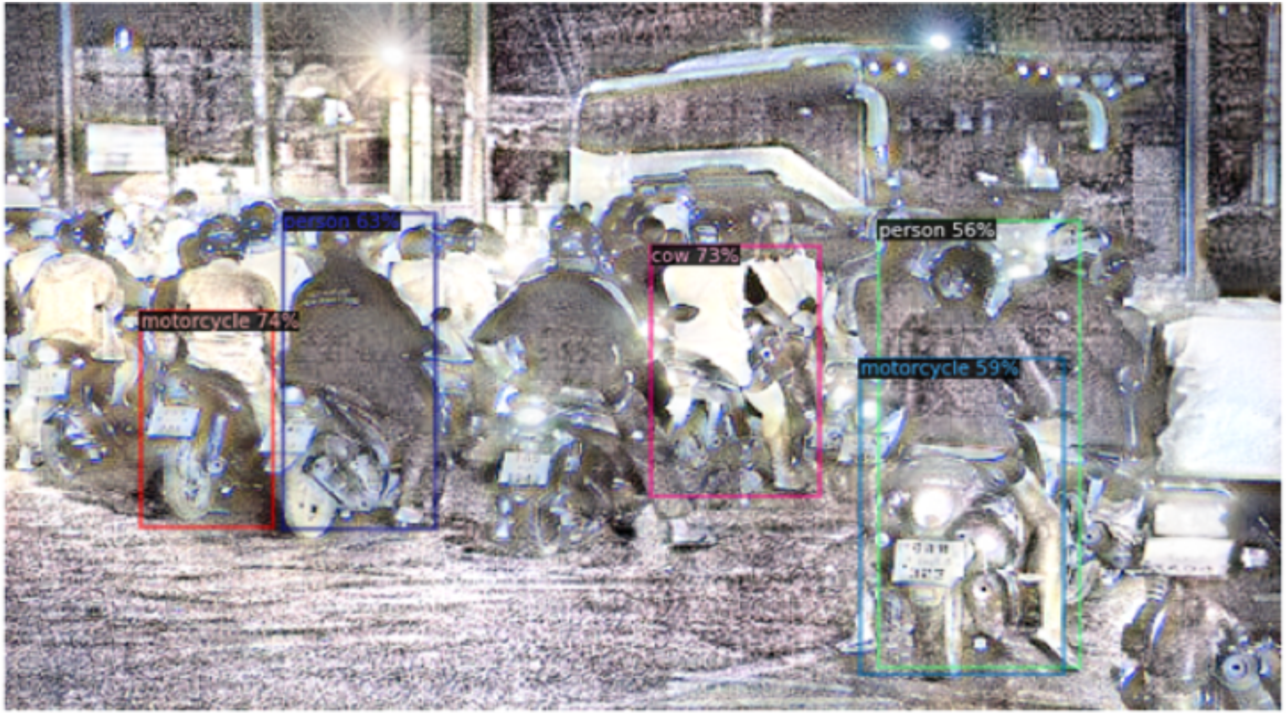
Contrast Limited Adaptive Histogram Equalization

CLAHE computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. Hence it is suitable for improving the local contrast and enhancing the definitions of the edges in each section of the image.





Architecture Diagram 3: Output



Architecture Diagram 4:

Architecture Diagram 4: Output

Architecture Diagram 5:

USM

Unsharp Mask

USM is an image sharpening technique.

Unsharp mask is a filter that amplifies the high-frequency components of an image. So the resulting image is less blurry than the original image with higher contrast and brightness.

Architecture Diagram 5: Output

Architecture Diagram 6:

Implementation Details of Finetuned F-RCNN

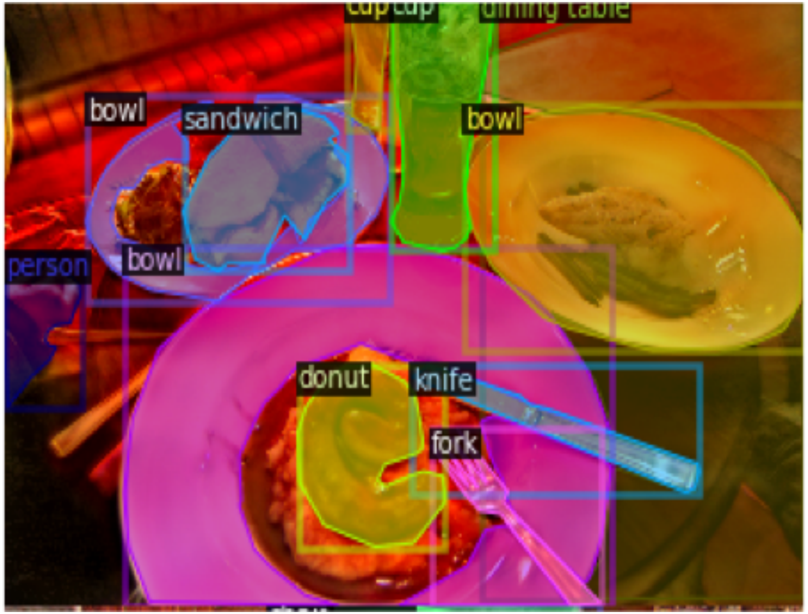
Register the dataset along with the annotations using Detectron

```
from detectron2.data.datasets import register_coco_instances
register_coco_instances("COCO_train_data", {}, "/content/drive/My Drive/COCO_annotations/instances_val2017_temp.JSON
```

```
from detectron2.data import MetadataCatalog
COCO_train_metadata = MetadataCatalog.get("COCO_train_data")
```

```
from detectron2.data import DatasetCatalog
dataset_dicts = DatasetCatalog.get("COCO_train_data")
```

Viewing sample images to verify annotations



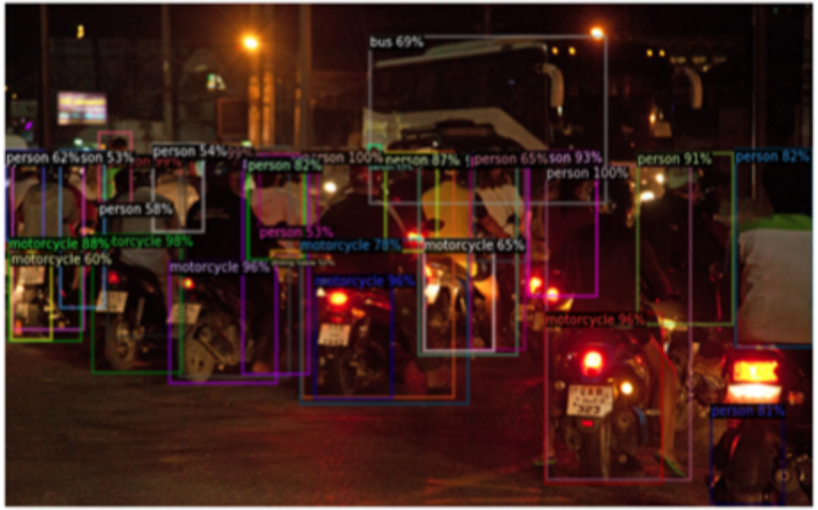
Fine-tuning Faster-RCNN on enlightened COCO images

```
import os
import numpy as np
import json
from detectron2.structures import BoxMode
import itertools
from detectron2.engine import DefaultTrainer
from detectron2.config import get_cfg

cfg = get_cfg()
cfg.merge_from_file("./detectron2_repo/configs/COCO-Detection/faster_rcnn_R_50_FPN_3x.yaml")
cfg.DATASETS.TRAIN = ("COCO_train",)
cfg.DATASETS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = "detectron2://COCO-Detection/faster_rcnn_R_50_FPN_3x/137849458/model_final_280758.pkl"  # initia
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 300
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 80

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()
```

Count of all the object categories

category	#instances	category	#instances	category	#instances
person	10777	bicycle	314	car	1918
motorcycle	367	airplane	143	bus	283
train	190	truck	414	boat	424
traffic light	634	fire hydrant	101	stop sign	75
parking meter	60	bench	411	bird	427
cat	202	dog	218	horse	272
sheep	354	cow	372	elephant	252
bear	71	zebra	266	giraffe	232
backpack	371	umbrella	407	handbag	540
tie	252	suitcase	299	frisbee	115
skis	241	snowboard	69	sports ball	260
kite	327	baseball bat	145	baseball gl..	148
skateboard	179	surfboard	267	tennis racket	225
bottle	1013	wine glass	341	cup	895
fork	215	knife	325	spoon	253
bowl	623	banana	370	apple	236
sandwich	177	orange	285	broccoli	312
carrot	365	hot dog	125	pizza	284
donut	328	cake	310	chair	1771
couch	261	potted plant	342	bed	163
dining table	695	toilet	179	tv	288
laptop	231	mouse	106	remote	283
keyboard	153	cell phone	262	microwave	55
oven	143	toaster	9	sink	225
refrigerator	126	book	1129	clock	267
vase	274	scissors	36	teddy bear	190
hair drier	11	toothbrush	57		
total	36335				

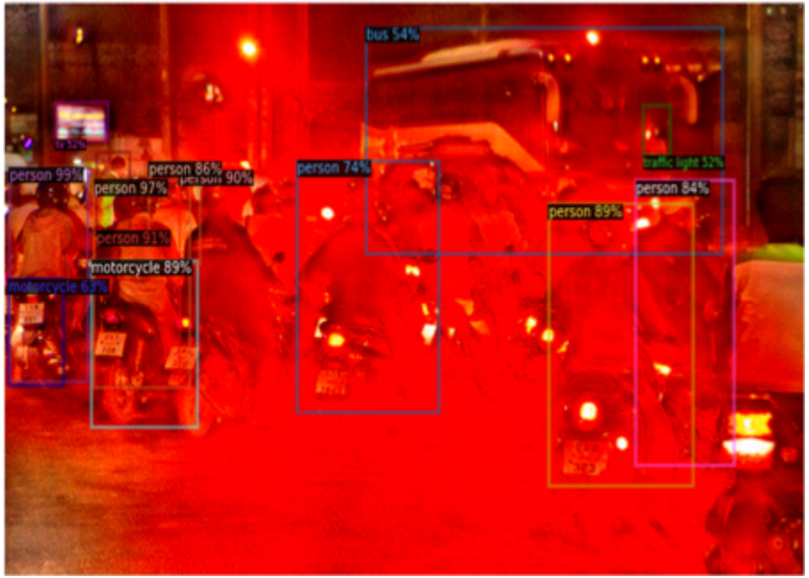
Results: Model 1



Faster R-CNN

Fine-tuned Faster R-CNN

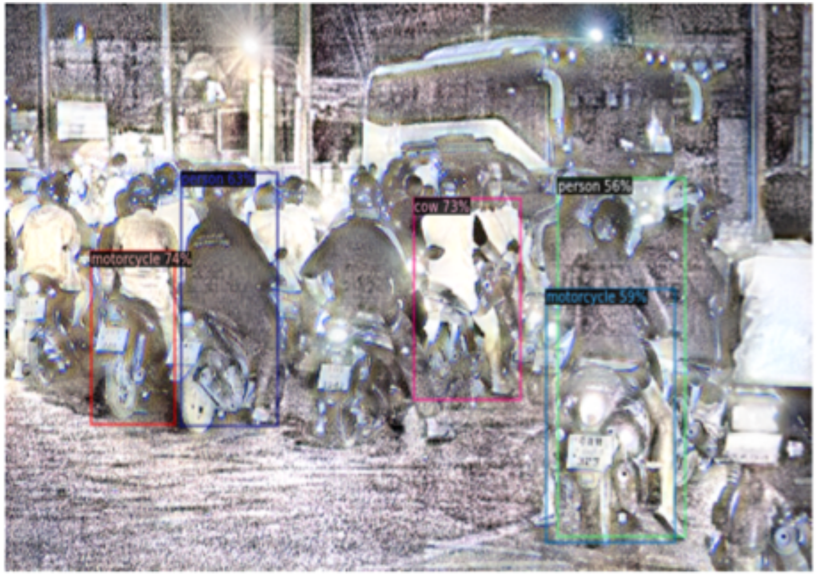
Results: Model 2



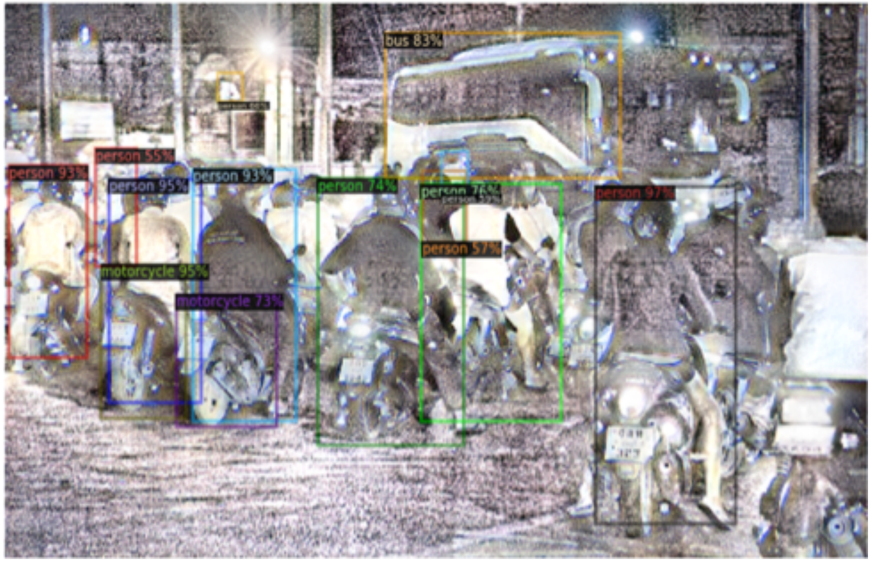
Faster R-CNN

Fine-tuned Faster R-CNN

Results: Model 3



Faster R-CNN



Fine-tuned Faster R-CNN

Results: Model 4

Faster R-CNN

Fine-tuned Faster R-CNN

Results: Model 5

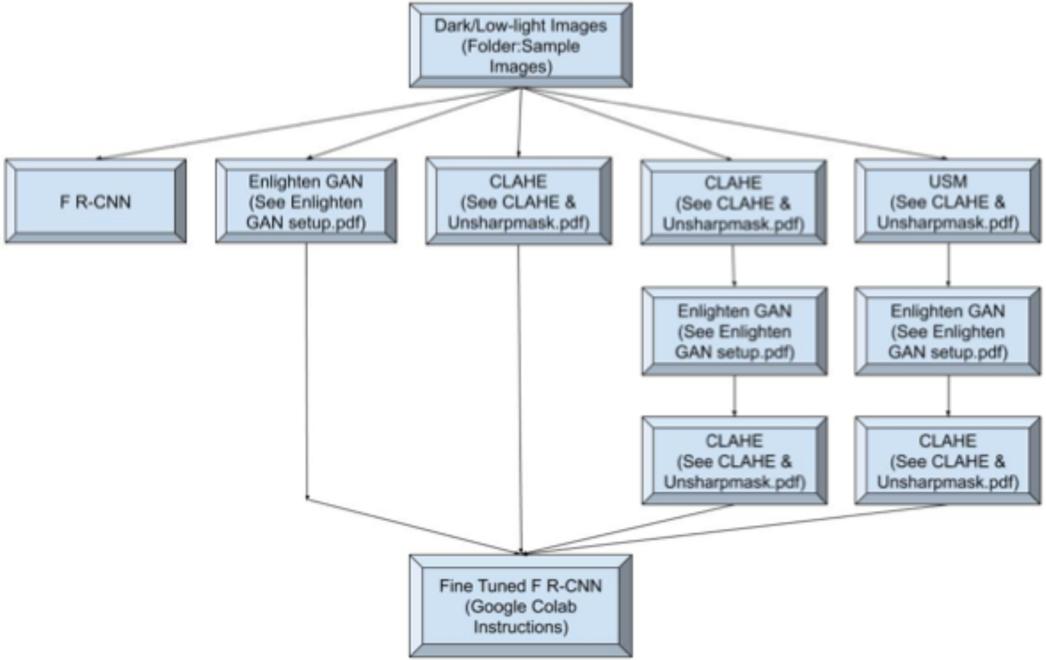
Results

Table 1. **Accuracy Comparison Table**

Model	Faster R-CNN Accuracy	Fine-tuned Faster R-CNN Accuracy
1	55%	72%
2	33%	**75%**
3	11%	33%
4	27%	41.66%
5	50%	52.77%

GitHub Implementation

Our program can be run by following the flowchart below



Conclusion

Extensive research work has been conducted on human and object detection for bright images.

The domain of human and object detection in dark images is majorly unexplored.

So the goal of this project is to evaluate the performance of Faster R-CNN by incorporating it with a combination of different image enhancement filters and convolutional neural networks.

We have fine Tuned Faster R-CNN that gives us the best accuracy for dark images in combination with enlightened GAN

Future Work

We are able to get a good accuracy for our fine tuned Faster R-CNN model, but we know that there is more to discover and work on.

For our future work, we would like to fine-tune Faster R-CNN with the Exclusively Dark dataset as showcased in the architecture diagram below and contribute to increasing the accuracy of human and object detection in the dark.

References

[1] Yuen Peng Loh, Chee Seng Chan, “Getting to Know Low-light Images with the Exclusively Dark Dataset”, 29 May 2018.
[2] Paul Viola, Michael Jones, “Robust Real-time Object Detection”, 13 July 2001.
[3] Thattapon Surasak, Ito Takahiro, Cheng-husan Cheng, Chi-en Wang, Pao-you Sheng, “Histo- gram of Oriented Gradients for Human Detection in Video”, 2018 5th International Conference on Business and Industrial Research (ICBIR), IEEE.
[4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, 6 Jan 2016.
[5] Niall O’ Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Deniel Riordan, Joeseph Walsh, “Deep Learning vs. Traditional Computer Vision”, Computer Vision Conference (CVC), 2019.
[6] Yifan Jiang, Xinyu Gong, Ding Liu, Yu Cheng, Chen Fang, Xiaohui Shen, Jianchao Yang, Pan Zhou, and Zhangyang Wang, “EnlightenGAN: Deep Light Enhancement without Paired Supervi- sion”.
[7] Victor T. Tom, Gregory J. Wolfe, “Adaptive Histogram Equalization And Its Applications”, 26th Annual Technical Symposium, 17 March 1983.
[8] Artyom M. Grigoryan, Sos S. Again, “Unsharp Masking”, Advances in Imaging and Electron Physics, 2004.
[9] Jindong Wang, Yiqiang Chen, Han Yu, Meiyu Huang, Qiang Yang; ‘EASY TRANSFER LEARNING BY EXPLOITING INTRA-DOMAIN STRUCTURES’
[10] https://github.com/facebookresearch/detectron2 [11] http://cocodataset.org/#download

Questions?

THANK YOU