iMat Fashion

Agenda

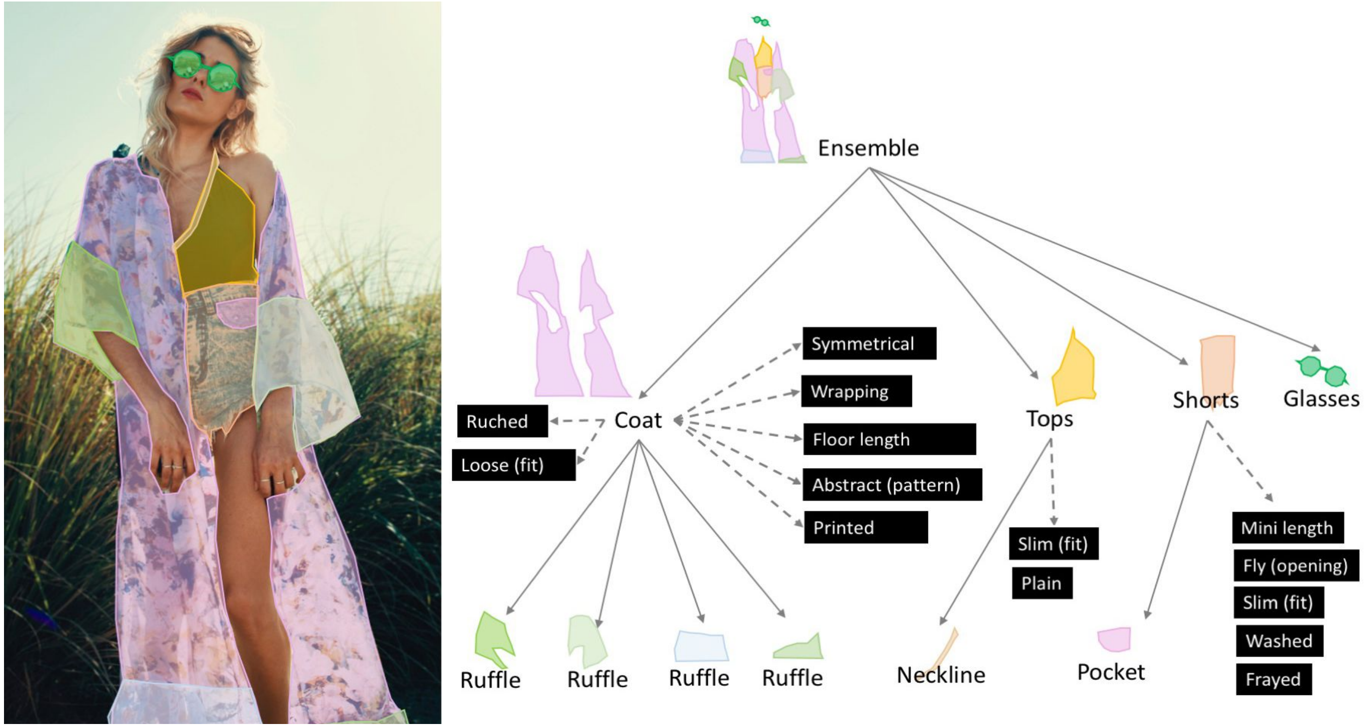
- Project Objective
- Data Overview & EDA
- Key Concepts
- Model Training
- Conclusion

Introduction

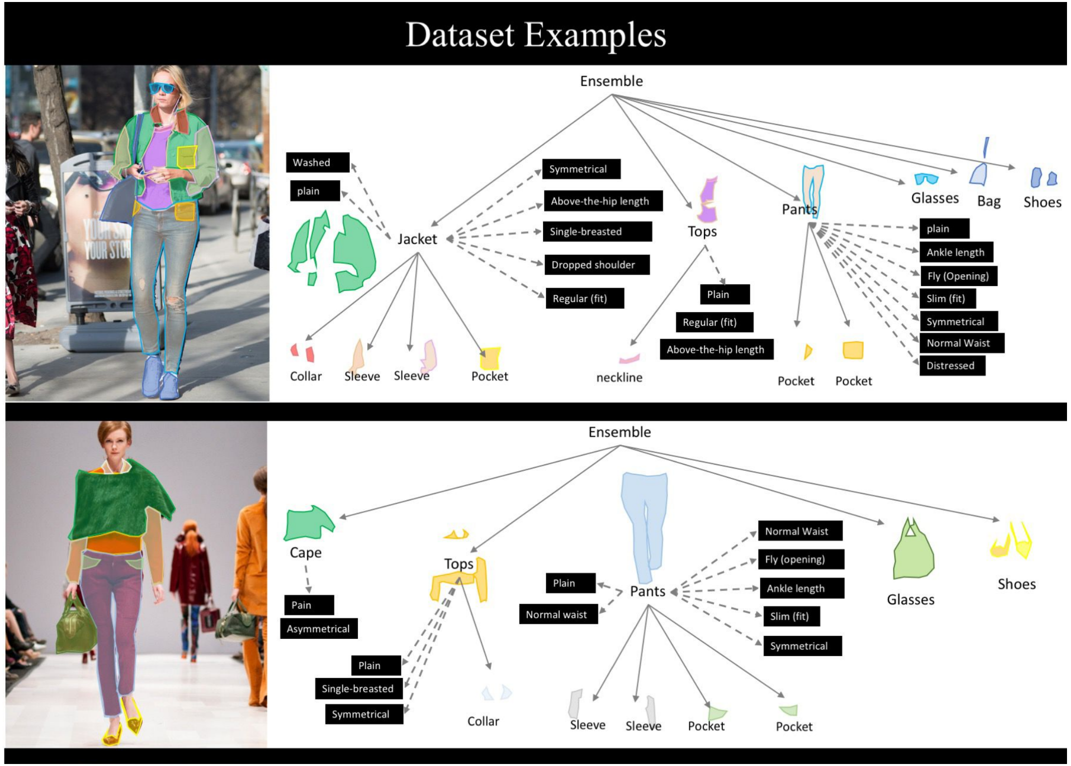
- Modern fashion is constantly evolving, making it nearly impossible for designers to keep up with current trends
- The future of fashion is about using data and adaptive visual analytics to discern consumer preference, to give the users what they want to look good
- Stated user preferences and global trends would help designers creating on trend, high value and exclusive designs

Objective

We are aiming to accurately assign attribute labels for any given fashion images and localize the pixels where the object is present.



Data Overview

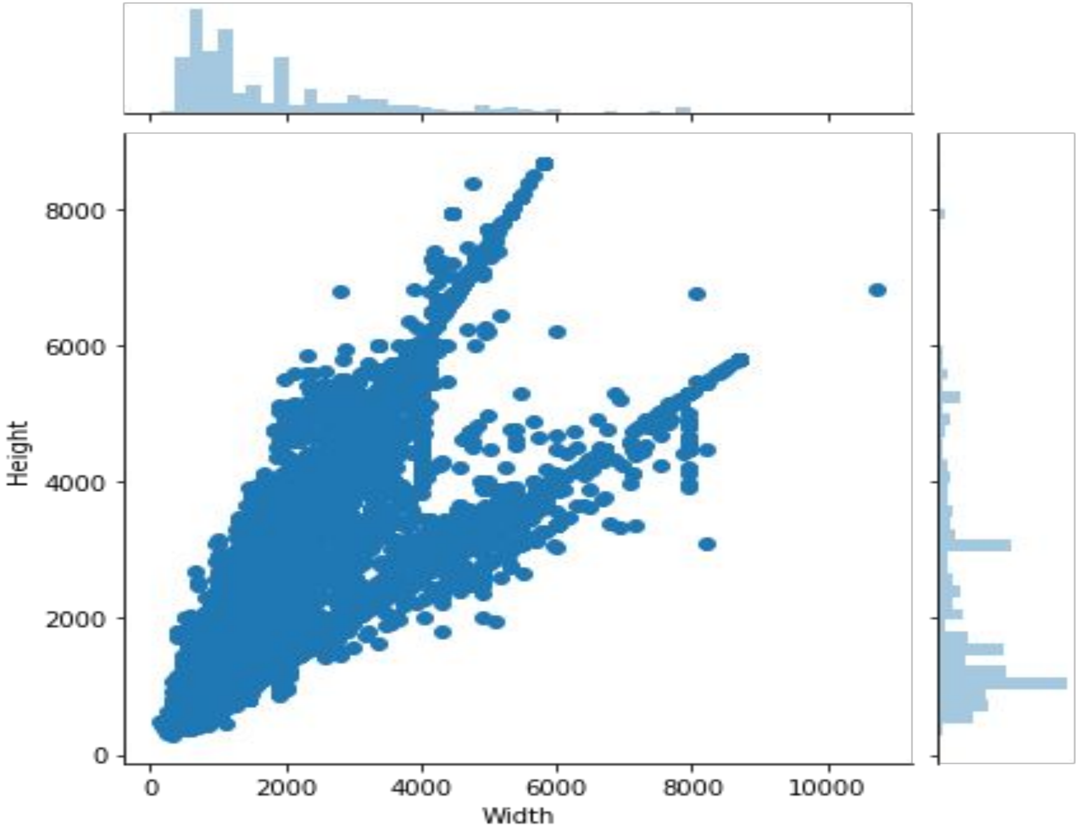


Kaggle Dataset

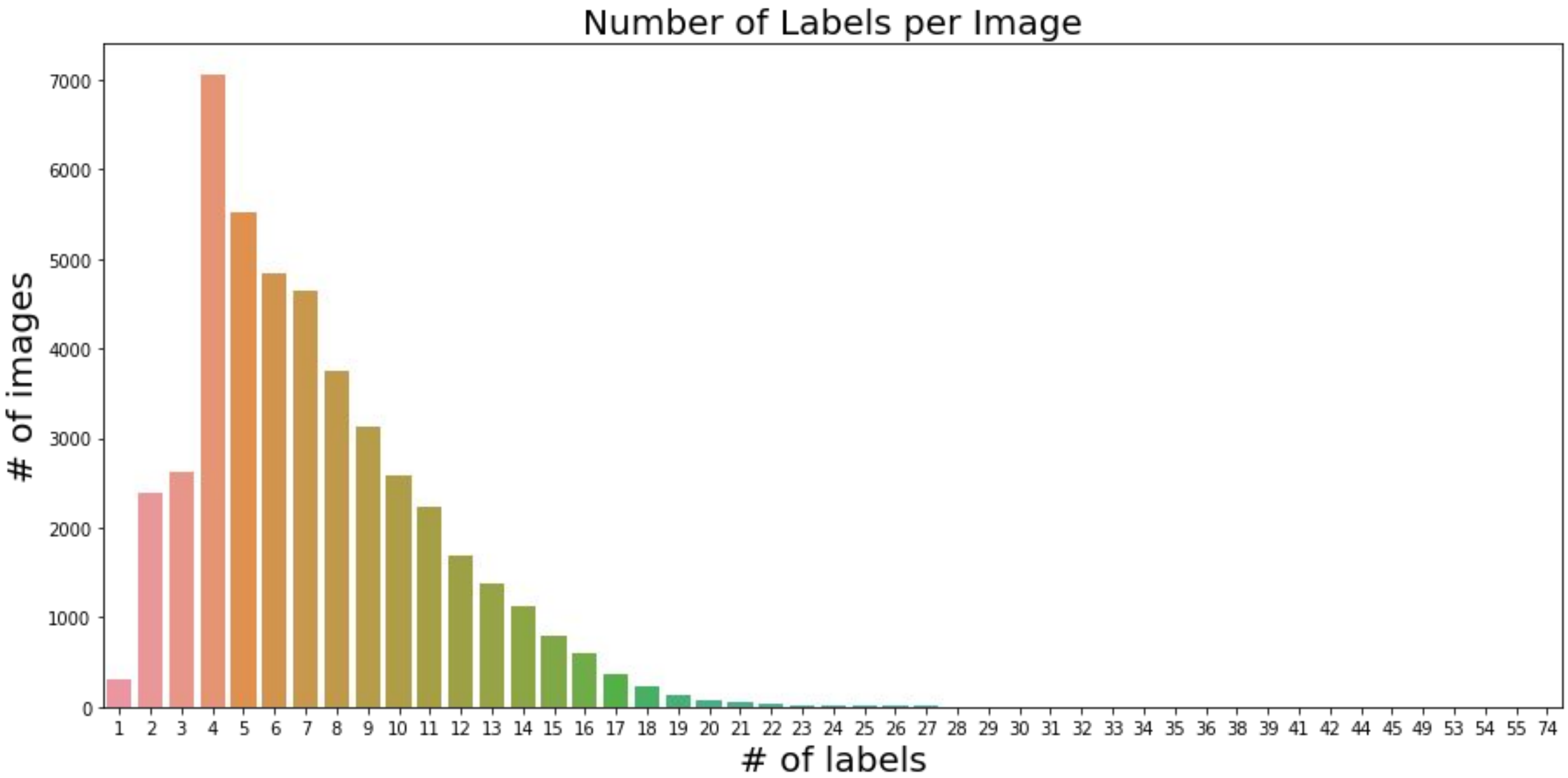
https://www.kaggle.com/c/imaterialist-fashion-2019-FGVC6

- 46 apparel objects
- 92 related fine-grained attributes
- 45,625 training records
- 3,200 test records

Image size distribution

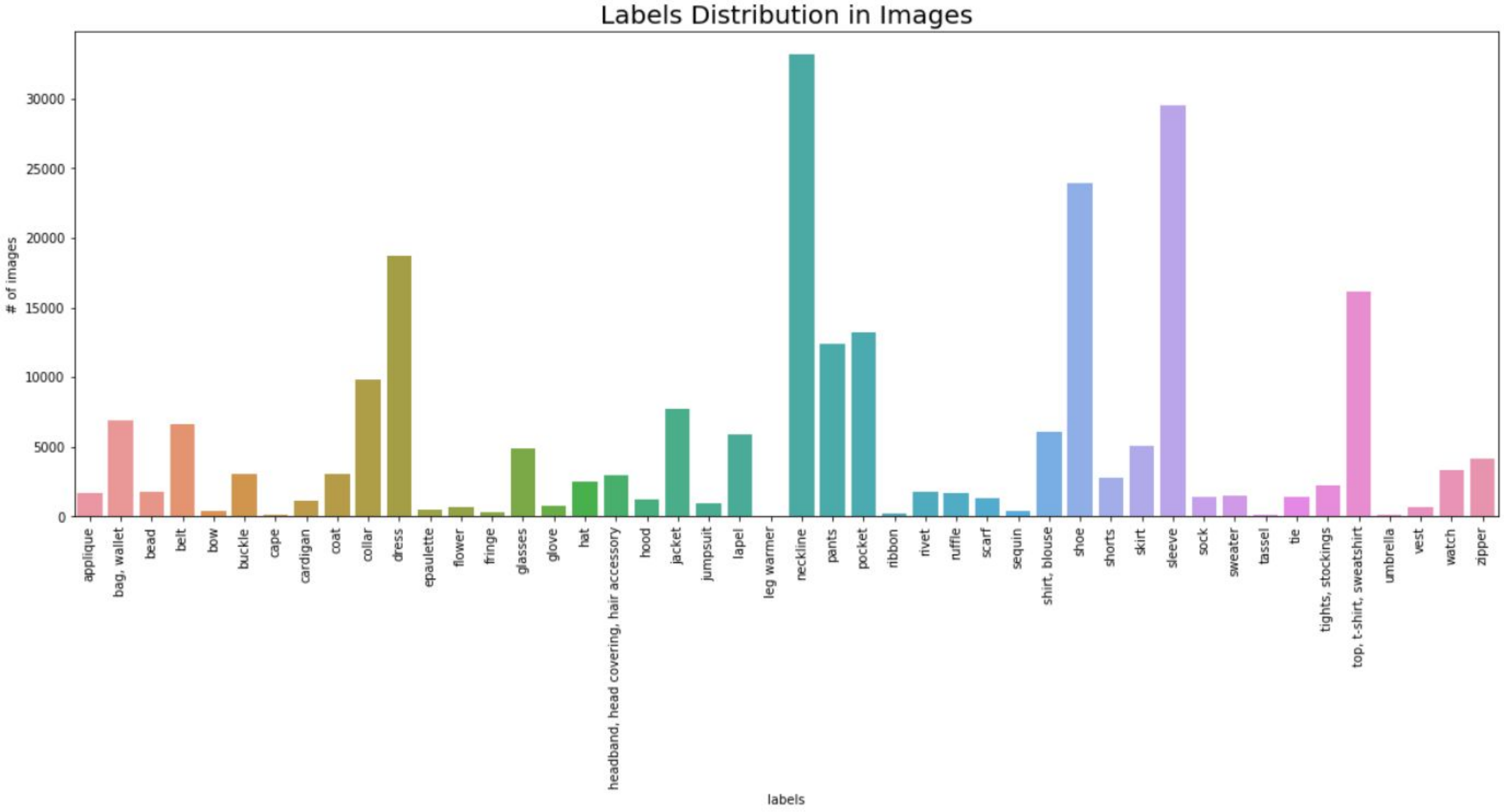


2/3rd of the images had between 2-10 labels

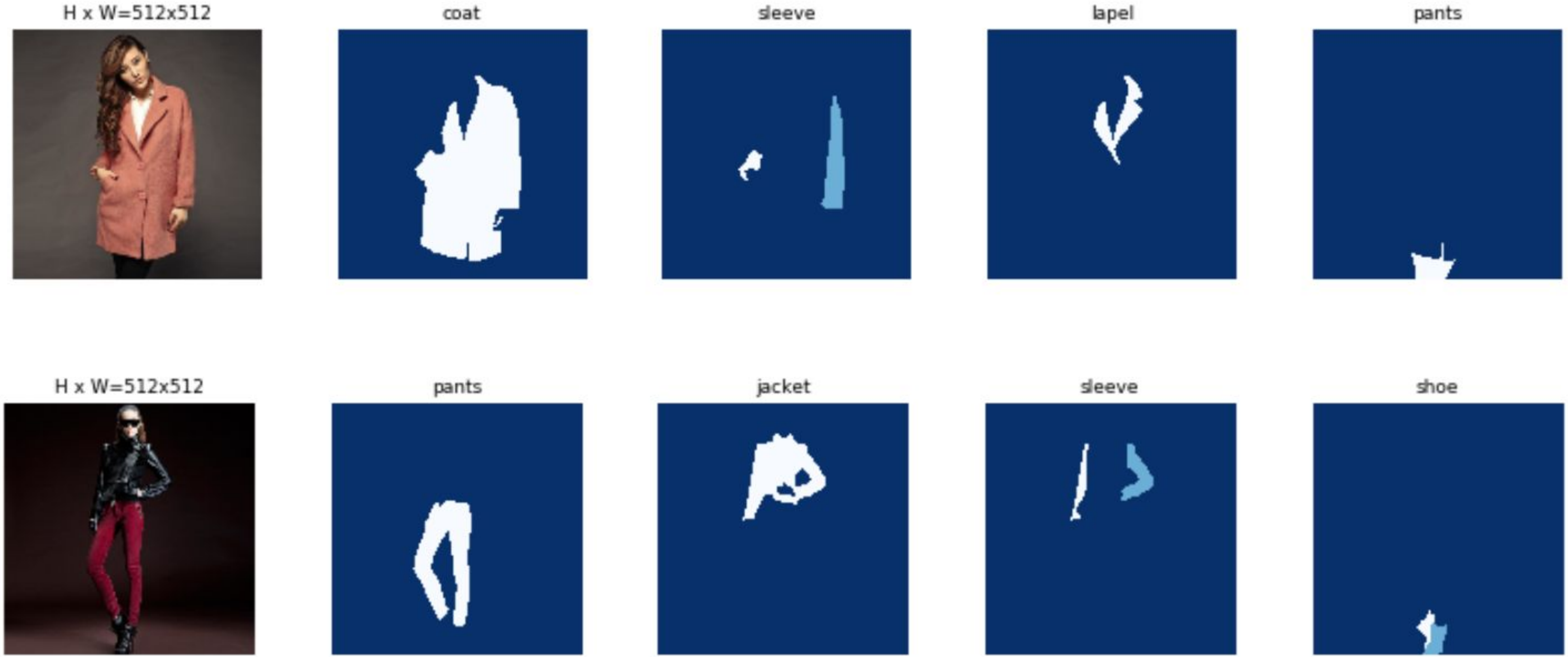


Labels in our data

Which are the most frequent label?

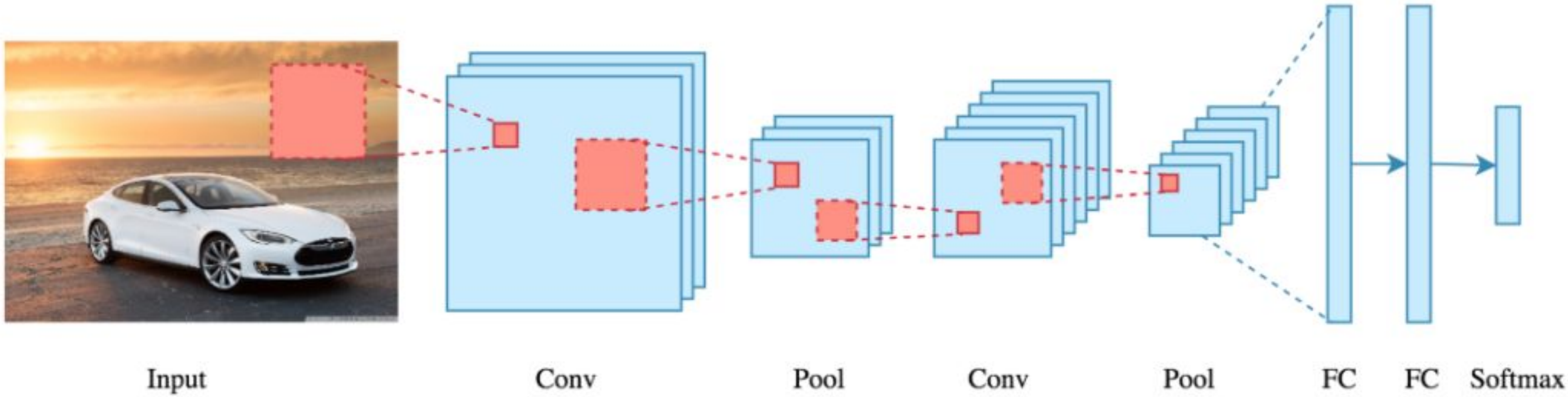


How does the training data look like?

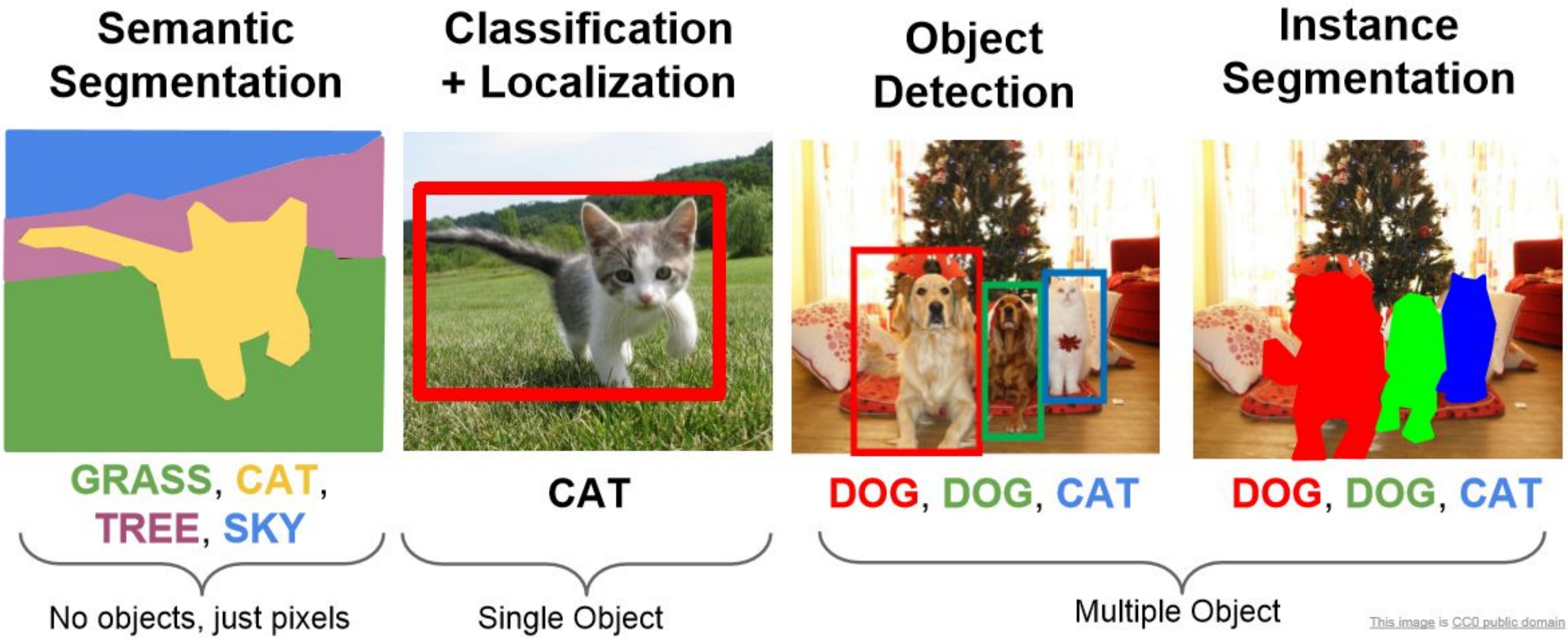


CNN in brief

All CNN models follow a similar architecture, as shown in the figure below.



Object detection



Challenges in Object Detection

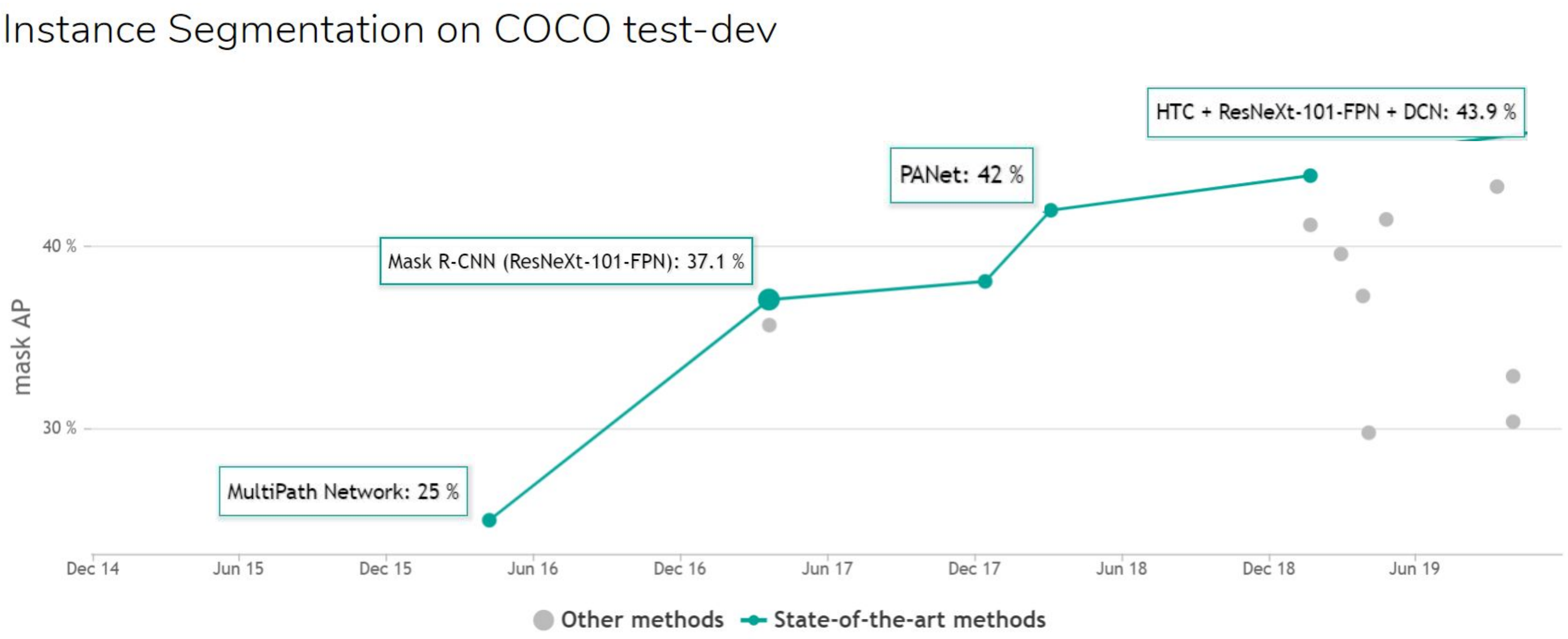
THE CHALLENGE:

- The number of objects
- Varying sizes and orientations of objects
- Traditional softmax loss and regression loss won't suit

SOLUTION: Split the image into smaller chunks and solve it as a classification and localization problem

HURDLES: What size to crop ? How many crops to consider ? What ratio ?

Timeline of development

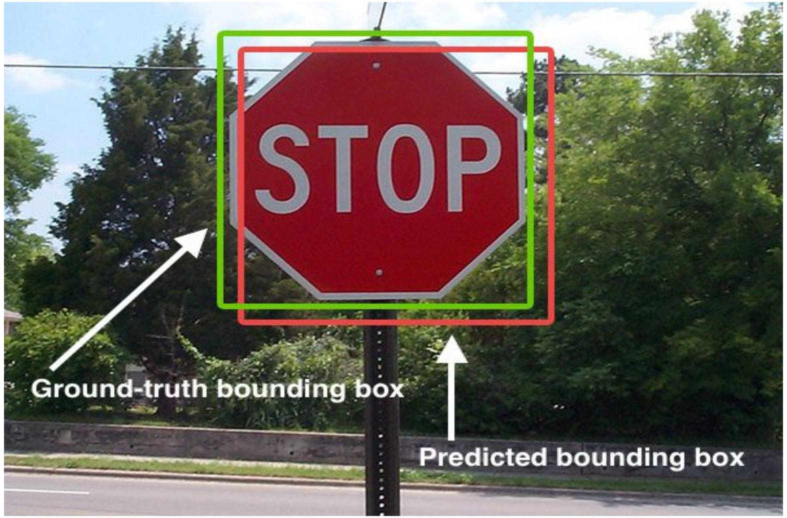


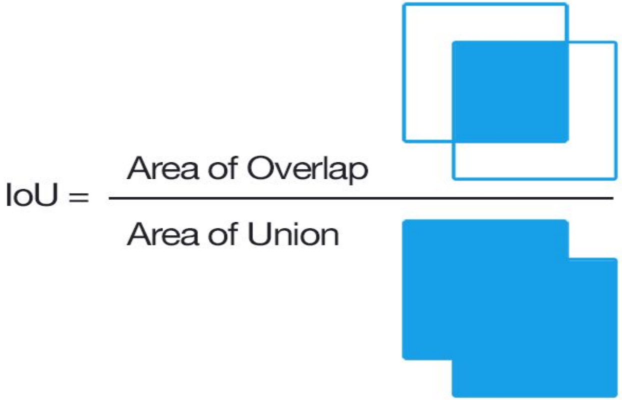
Source: https://paperswithcode.com/sota/instance-segmentation-on-coco

Our Evaluation Metric

- The metric sweeps over a range of IoU thresholds from 0.5 to 0.95 with a step size of 0.05

- At each threshold value, a precision value is calculated based on the difference between predicted and the ground truth objects





$$\frac{1}{|threshold|}\sum_{t}\frac{TP(t)}{TP(t)+FP(t)+FN(t)}$$

Understanding Hyperparameters

Start of Project

- Loss Weights:
  - "rpn_class_loss": 1.0
  - "rpn_bbox_loss": 1.0
  - "mrcnn_class_loss": 1.0
  - "mrcnn_bbox_loss": 1.0
  - "mrcnn_mask_loss": 1.0
- Back Bone:
  - ResNet101

Best Kaggle Score

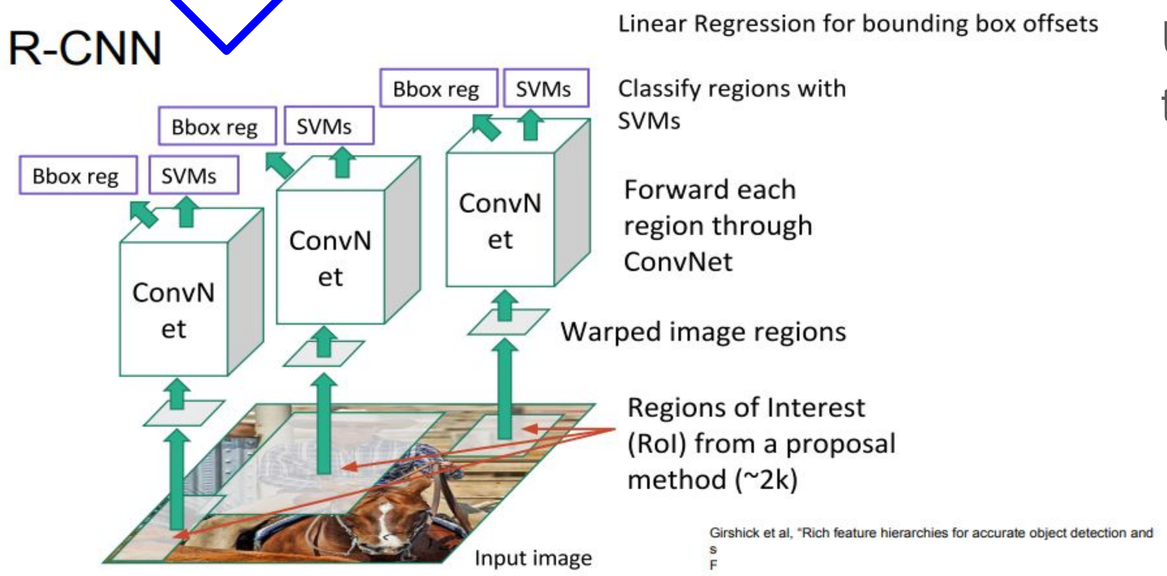
- Loss Weights:
  - "rpn_class_loss": 1.0
  - "rpn_bbox_loss": 0.8
  - "mrcnn_class_loss": 6.0
  - "mrcnn_bbox_loss": 6.0
  - "mrcnn_mask_loss": 6.0
- Back Bone:
  - ResNet50

Best for Identification

- Loss Weights:
  - "rpn_class_loss": 10.0
  - "rpn_bbox_loss": 0.8
  - "mrcnn_class_loss": 6.0
  - "mrcnn_bbox_loss": 6.0
  - "mrcnn_mask_loss": 6.0
- Back Bone:
  - ResNet50

Evolution of Mask R-CNN

- R-CNN: Use Region Proposals from Computer vision techniques
- Fast R-CNN: Use a conv Net before Region Proposals to reduce computation
- Faster R-CNN: Regional Proposal Network to propose ROI
- Mask R-CNN: Add a Conv net for Semantic Segmentation on Faster RCNN

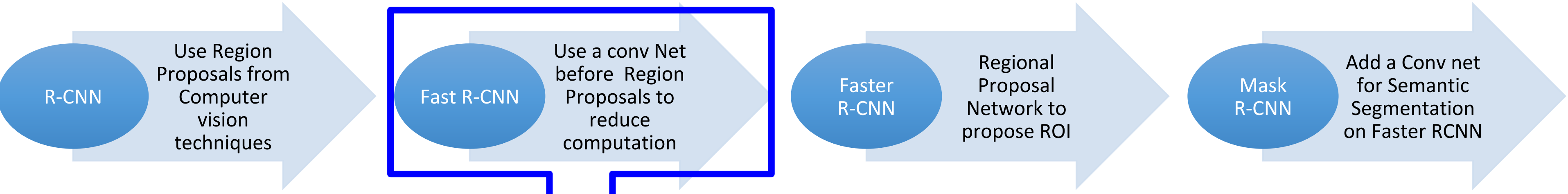


Use Region Proposals from CV techniques

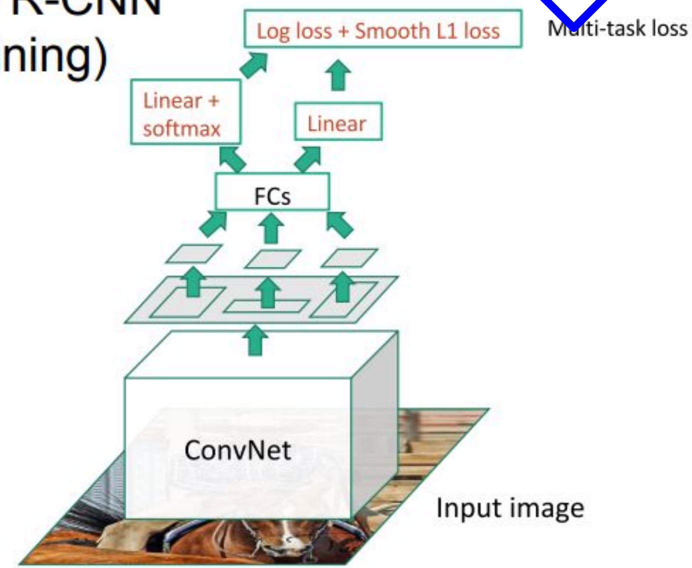
- Computationally expensive, Close to 2000 region proposals for each image
- Training is super slow
- Very High Disk Space

Image Source: Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

Evolution of Mask R-CNN



Fast R-CNN
(Training)



Advantages:

- Reduces the computation by shifting from multiple convnets to single convnet before ROI Pooling
- 49 Hours ~ 2.3 hours*
- Computing Region Proposals dominate the computation time

Image Source: Girshick, “Fast R-CNN”, ICCV 2015; source

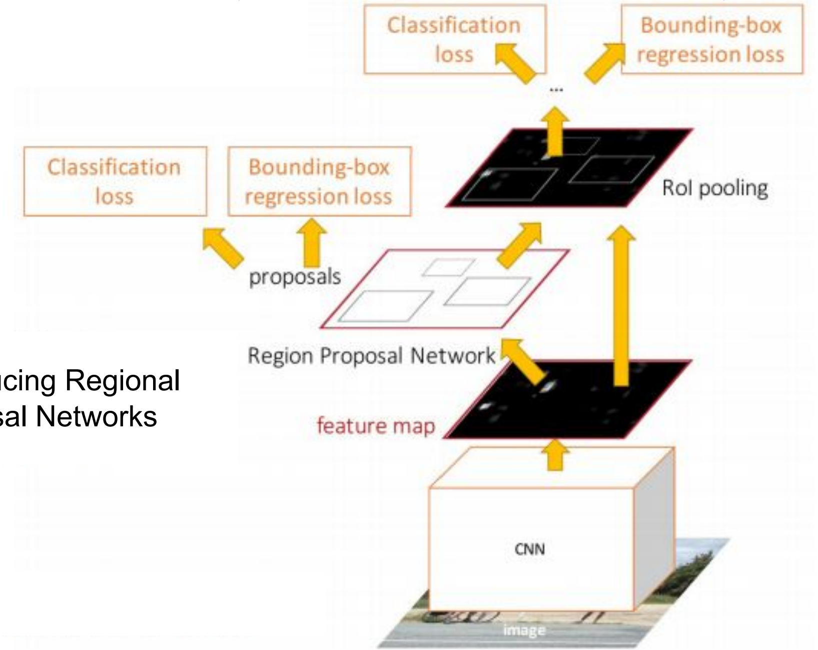
Evolution of Mask R-CNN

R-CNN: Use Region Proposals from Computer vision techniques

Fast R-CNN: Use a conv Net before Region Proposals to reduce computation

Faster R-CNN: Regional Proposal Network to propose ROI

Mask R-CNN: Add a Conv net for Semantic Segmentation on Faster RCNN



Introducing Regional Proposal Networks

- Use the Convnet - Region proposal Network to reduce the computation time
- Increasing Number of Losses to 4, 2 for Region Proposals and 2 for Classification and Bounding Box

Image Source: Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

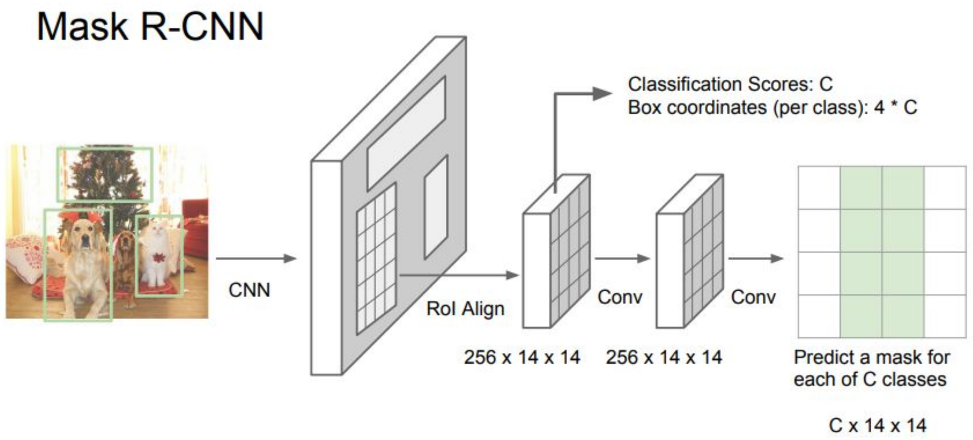
Evolution of Mask R-CNN

R-CNN: Use Region Proposals from Computer vision techniques

Fast R-CNN: Use a conv Net before Region Proposals to reduce computation

Faster R-CNN: Regional Proposal Network to propose ROI

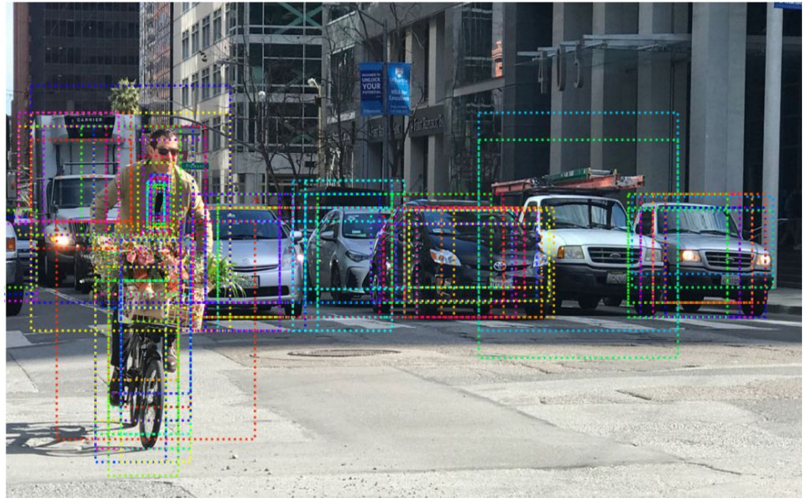
Mask R-CNN: Add a Conv net for Semantic Segmentation on Faster RCNN

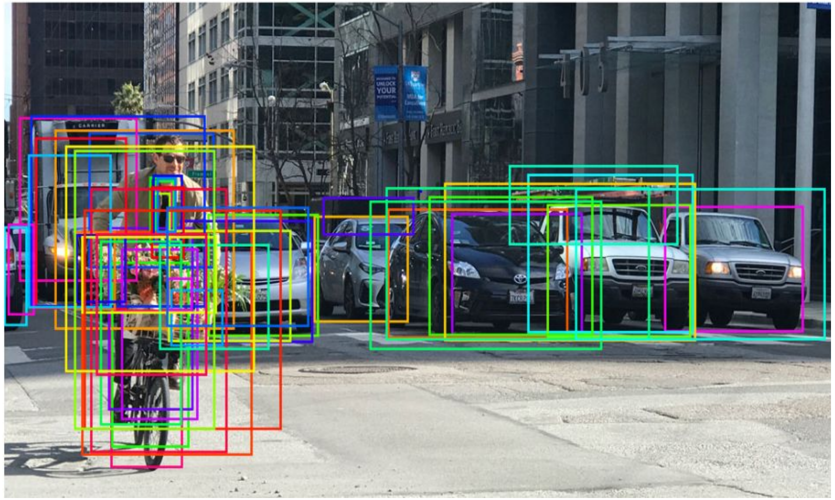


- Add conv layers for segmentation on the output of identified Faster R-CNN Classes
- Use FCN for prediction of Masks
- ROI align (without quantisation) used instead of ROI pool
- End - to End Process for instance Segmentation

Source: He et al, “Mask R-CNN”, arXiv 2017

Mask R-CNN : Steps - RPN

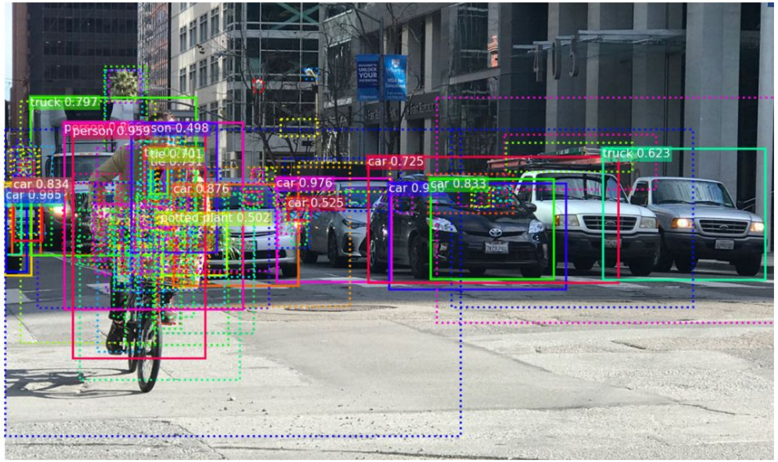


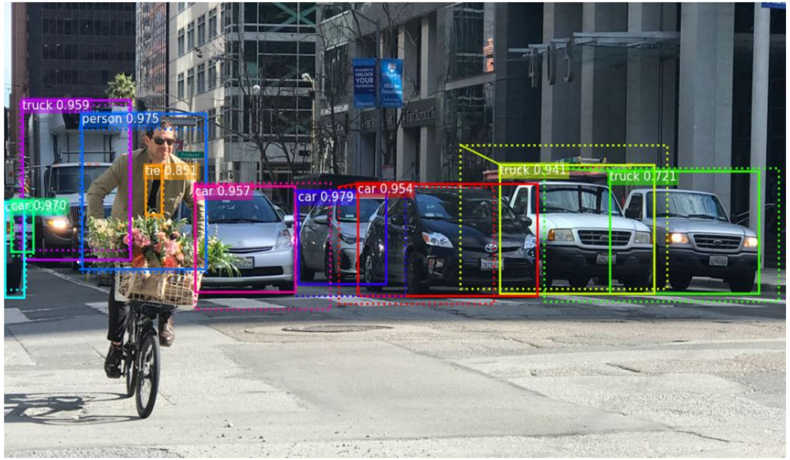


- Using the region proposal network, to make ROI proposals. The dotted rectangles below are those proposals
- Refine the boundary box better and identify the boundary box encloses the ground truth objects better.

Image Source: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

Mask R-CNN : Steps - Detection

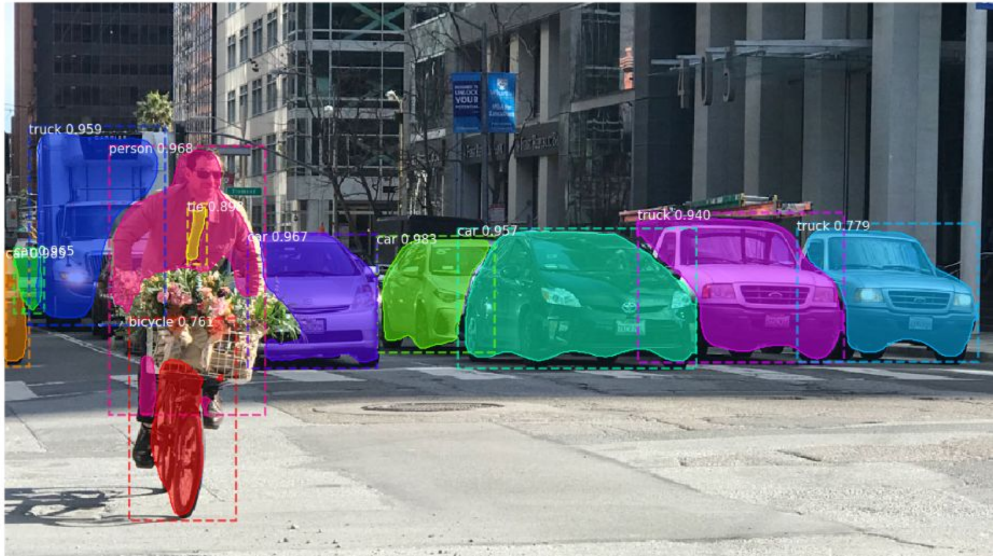




- Similar to Faster R-CNN, it performs object classification based on the ROIs (dotted lines) from RPN. The solid line is the boundary box refinements in the final predictions.
- Groups highly-overlapped boxes for the same class and selects the most confidence prediction only. This avoids duplicates for the same object

Image Source: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

Mask R-CNN : Masking



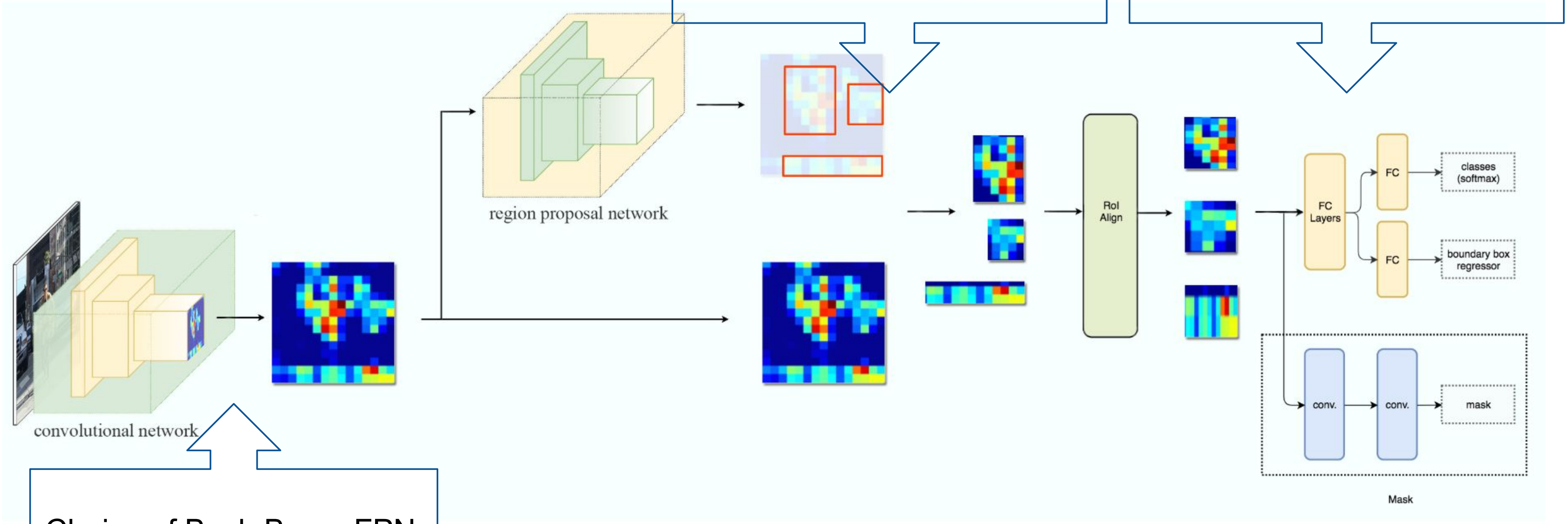
- Application of Mask to segment each of the instances identified and is the final Output of the Model

Image Source: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

Mask R-CNN Architecture Summary

RPN Class Loss, RPN BBox Loss

M-RCNN Class Loss, M-RCNN BBox Loss



Choice of Back-Bone, FPN

M-RCNN mask loss weight

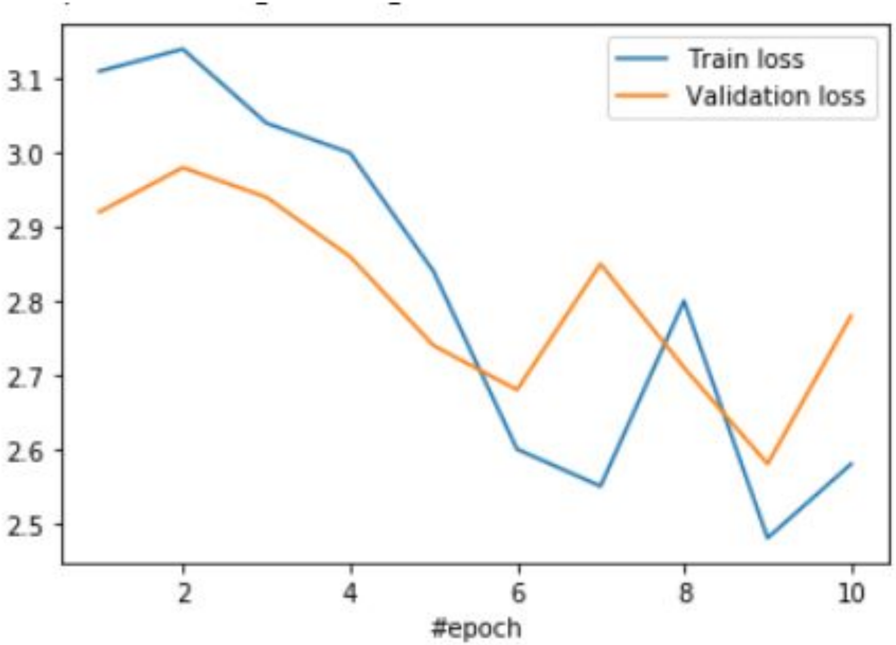
Image Source: https://medium.com/@jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

How we approached

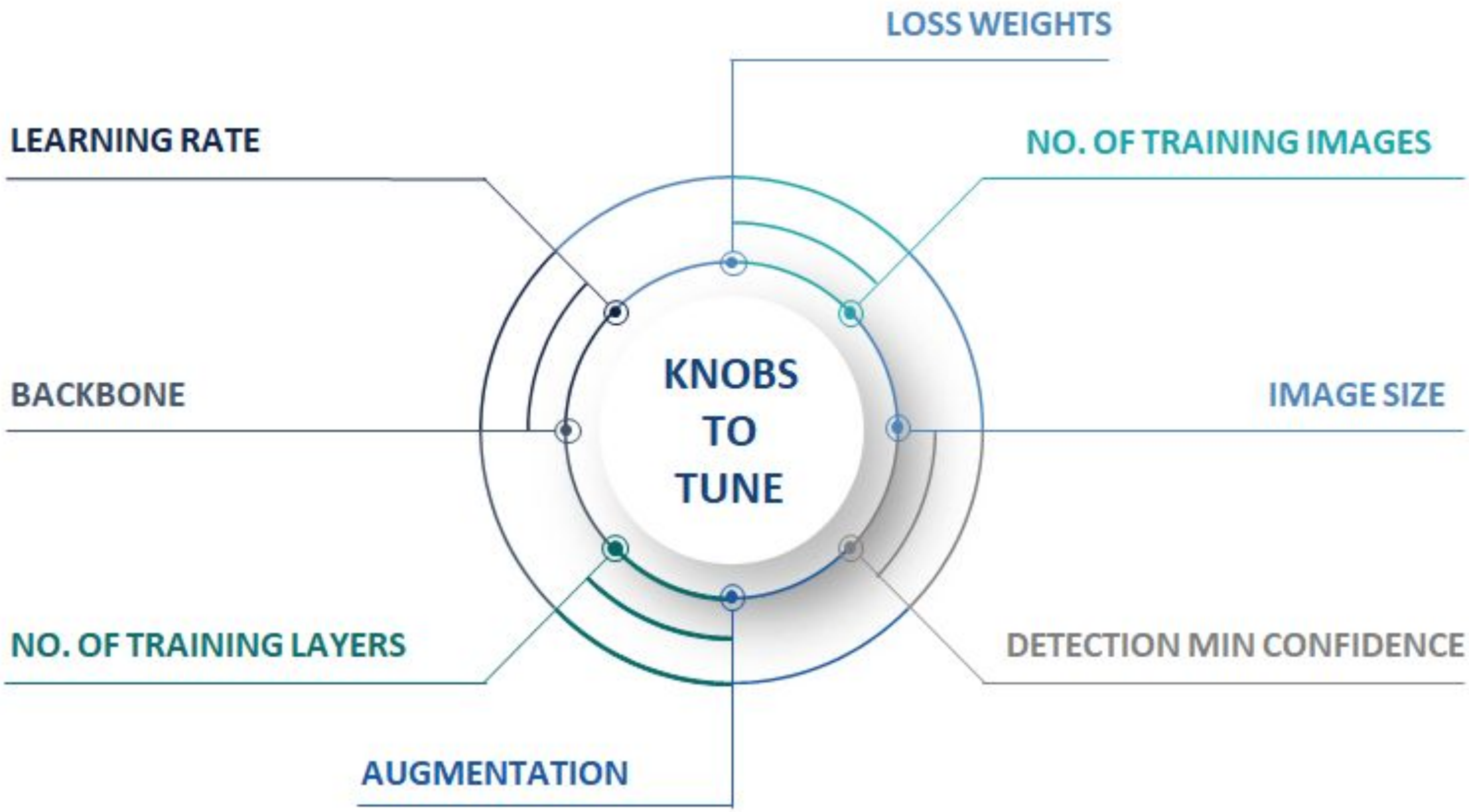
1. **Used Matterport's implementation of Mask R-CNN which is based on ResNet backbone**
2. Took a small sample of training set (~5000 images) to train the model using Google Colab
3. Leveraged weights trained on COCO dataset and trained all the layers to customize it for our problem
4. Tuned the model using different combinations of hyperparameters(learning rate, epochs, number of dense layers and nodes)

Transfer Learning and training the head layer

- Used Matterport's Mask R-CNN implementation
- COCO weights
- Resnet 50 backbone
- Weight Decay: 0.0001
- Momentum: 0.90
- Learning Rate: 0.001
- 10 epochs
- Test Score: 0.044

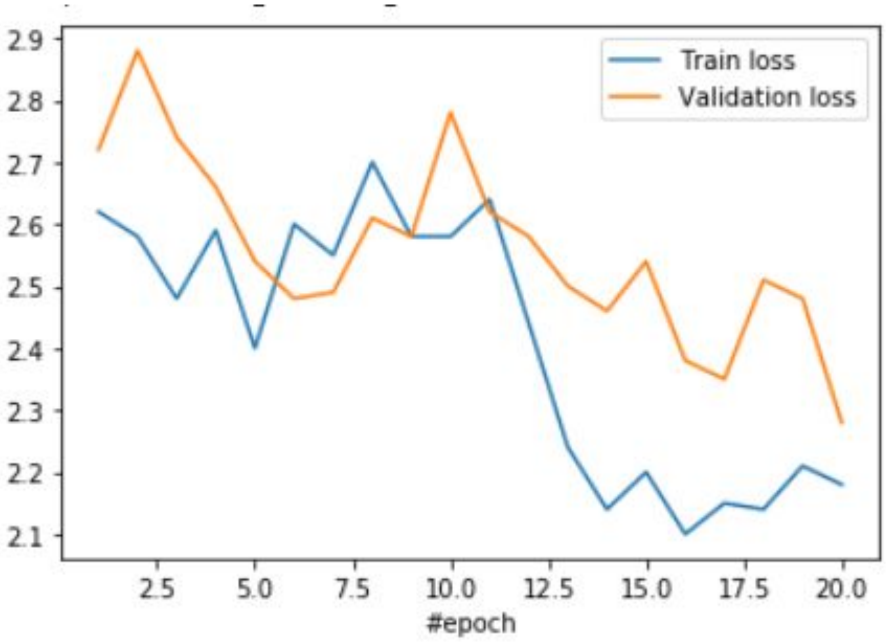


Approach - Knobs to tune



Model Training: All layers

- Trained all layers
- Learning Rate:
  - 10 epochs: 0.001
  - 20 epochs: 0.0001
- Test Score: 0.063

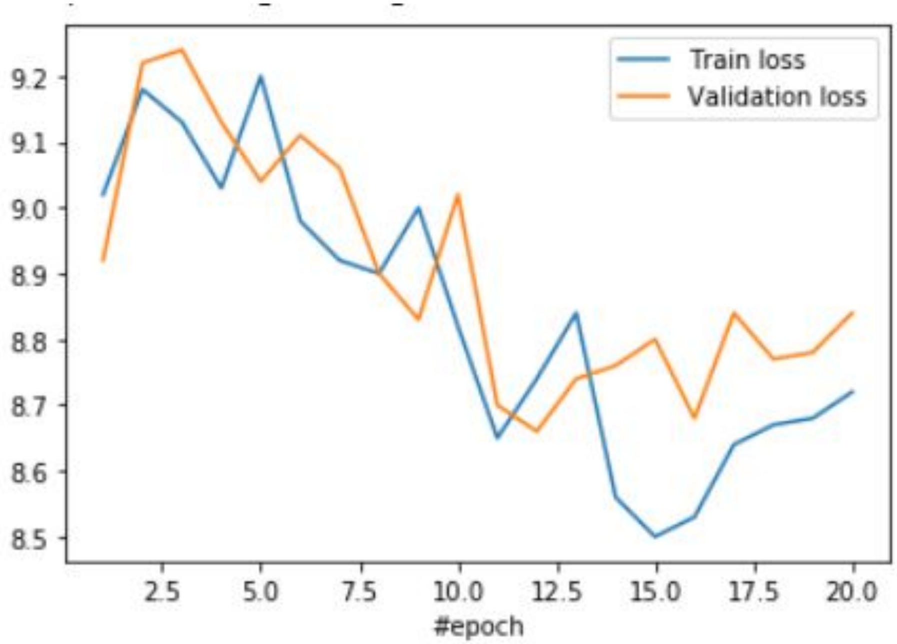


Improving the model performance be like

Model Training: Augmentation

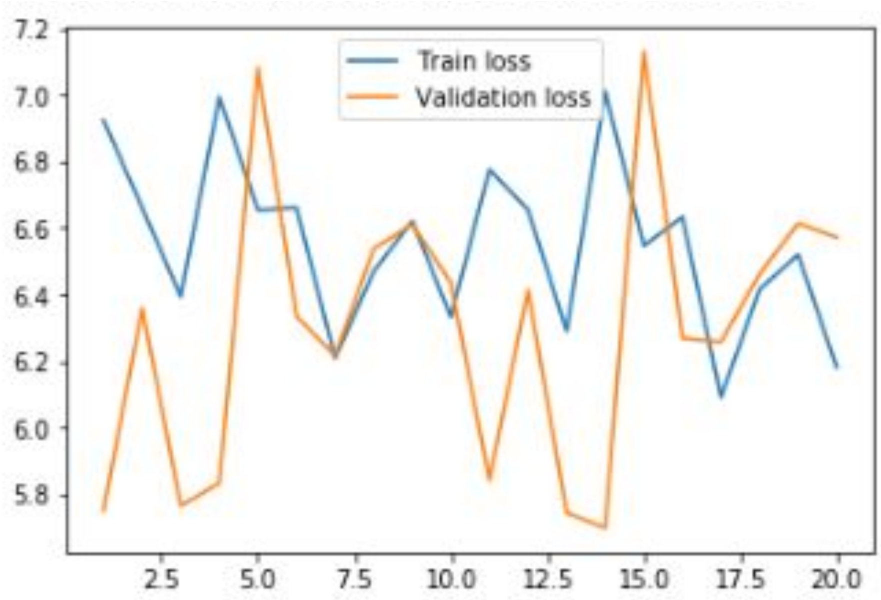
Model Training: Augmentation and Tweaked Loss Weight

- Trained all layers
- Learning Rate:
  - 10 epochs: 0.0001
  - 20 epochs: 0.00005
- Loss Weights:
  - "rpn_class_loss": 10.0
  - "rpn_bbox_loss": 0.8
  - "mrcnn_class_loss": 6.0
  - "mrcnn_bbox_loss": 6.0
  - "mrcnn_mask_loss": 6.0
- Test Score: 0.078

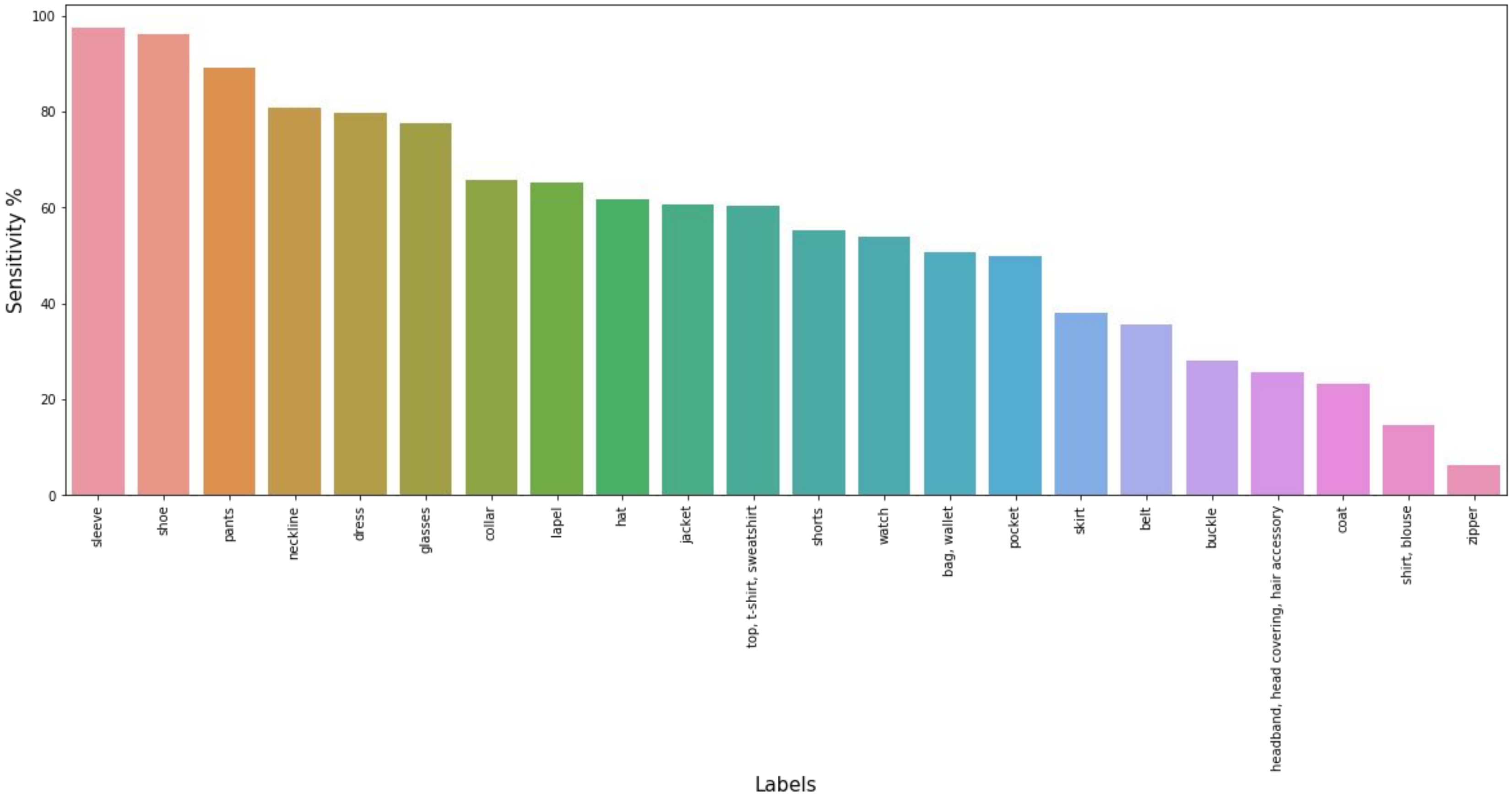


Model Training: Increased Training Images to 15k

- Trained all layers
- Learning Rate:
  - 10 epochs: 0.00005
  - 20 epochs: 0.00003
- Loss Weights:
  - "rpn_class_loss": 1.0
  - "rpn_bbox_loss": 0.8
  - "mrcnn_class_loss": 6.0
  - "mrcnn_bbox_loss": 6.0
  - "mrcnn_mask_loss": 6.0
- Detection max instances= 50
- Test Score: 0.081



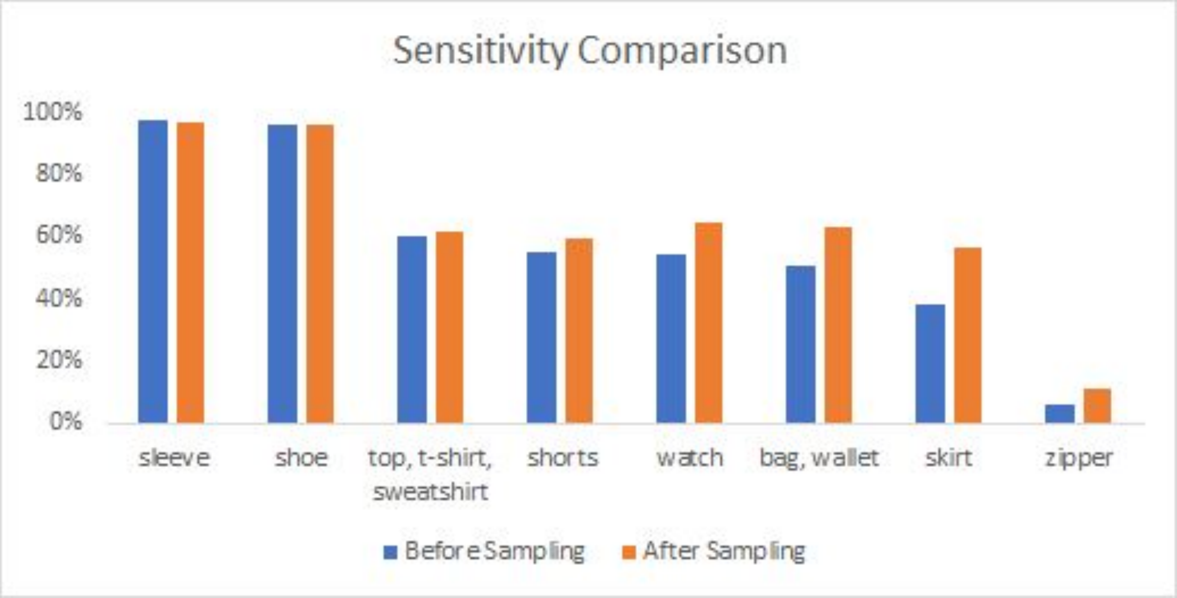
Model Training: What are we missing?



Model Training: Chasing the error!

Resampled training data to include more images for the 5 most occurring but least sensitivity score classes

- # Images: 3000
- Learning Rate:
  - 5 epochs: 0.0001
  - 10 epochs: 000005
- Test Score: 0.082



Results

True Labels - Top-Tshirt, Shorts, Sleeve, Collar, Belt, Sock, Shoe

Next Steps: Come Join our Thanksgiving Celebration!!!

- Using optimized parameters, train on all 45k images
- Try smooth resizing
- Try integrating classification algorithms to filter Mask RCNN outputs with confidence
- Try deploying the final model as a web service

Possible Industry Applications

- **Apparel Search through Phone App:** Working on the concepts of Google Lens - Shoppers can search for products using their phone camera
- **Fast-fashion Trend Analysis:** Retailers can study emerging trends in fashion and host them in their product assortment before anyone else
- **Product Recommendation Engine:** Can help retailers recommend to their shoppers, products with similar attributes to the one they're looking at. For example, they can recommend alternatives to out-of-stock products, so customers don't bounce off their website easily

THANK YOU!

Image size distribution