Simpsons Character Recognition Project

Siqi Li, Lirong Ma, Zhangyi (Rocky) Ye, Haonan Zhang,
Yuxuan (Nancy) Zhang, Luping (Rachel) Zhao

The Simpsons

Popular American animated sitcom

On air since 1989

Problem Statement

Problem: Large number of characters, sometimes we don’t know who characters are while watching the show.

Goal: build deep learning models to detect and classify Simpsons characters.

Deployment:

Real-time character detection and classification.

Viewers know who they are watching without pressing pause to check their phones.

Dataset

Public dataset

18,992 pictures for 18 characters

Divide into 3 sets of data

- Training (60%)
- Validation(20%)
- Test (20%)

Dataset Examples - Easy

Dataset Examples - Medium

Dataset Examples - Hard

Dataset - Challenge

Wrong character labels

Missing bounding box labels for images with multiple characters

- Affect our ability to test model performance for part 2 detection

Technical Approach

Data Cleaning & Augmentation

Convert images to pixels and normalized

Randomly rotate images (rotationrange=15)

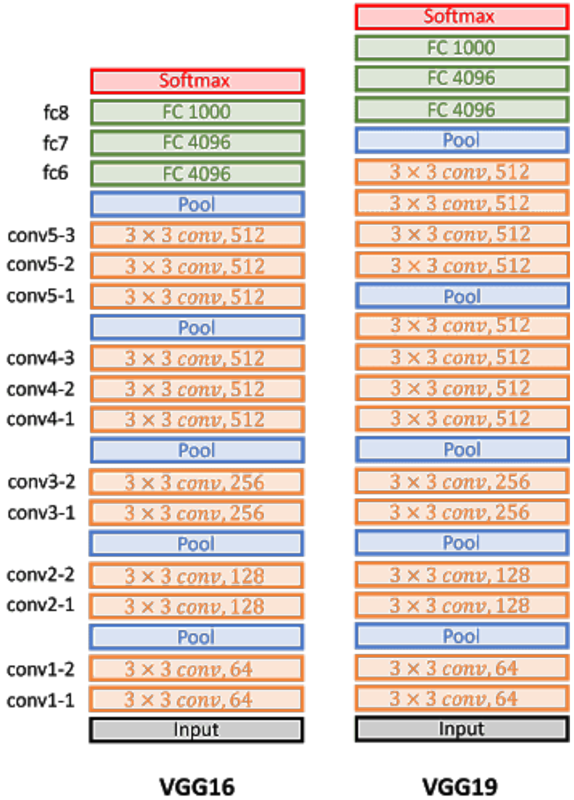
Randomly zoom inside pictures (zoom_range=0.2)

Randomly apply shearing transformations (shear_range=0.2)

Randomly flip images horizontally

Randomly shift images horizontally/vertically (shift_range=0.2)

VGG16/VGG19



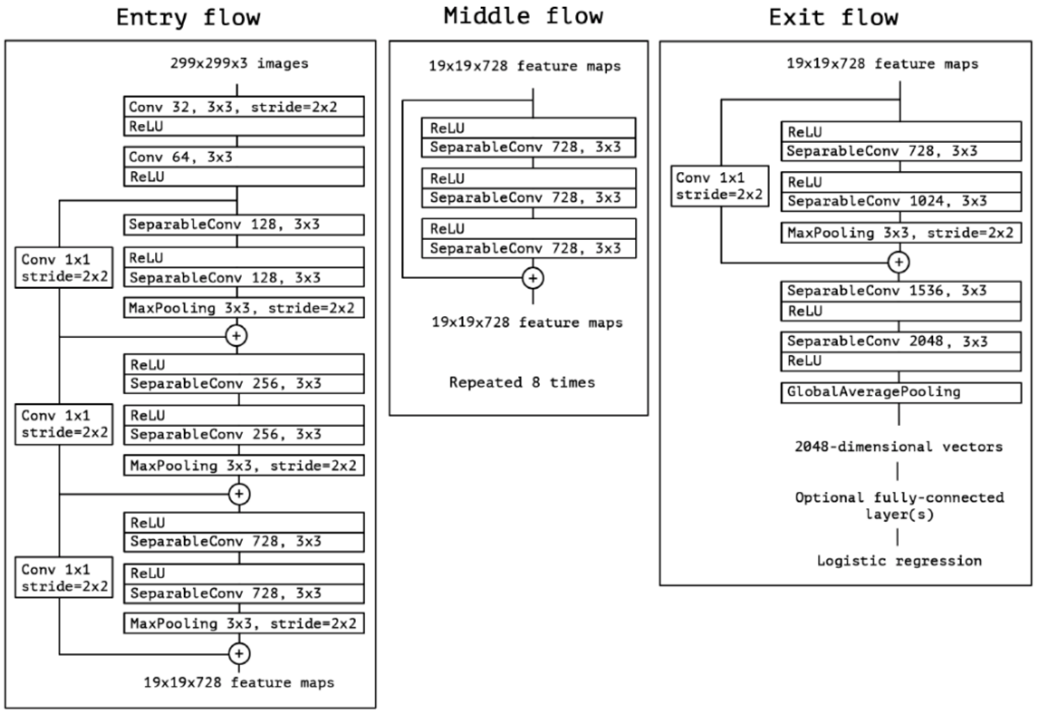
VGG16

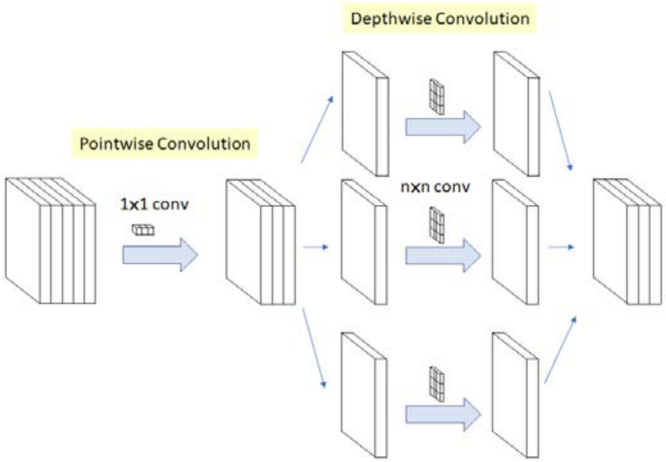
- Convolutional layers
- Pooling layer
- Flatten layer
- Dropout layer

VGG19

- Similar to VGG16, except it has four convolutional layers in the fourth and fifth block.

Xception (Extreme version of Inception)





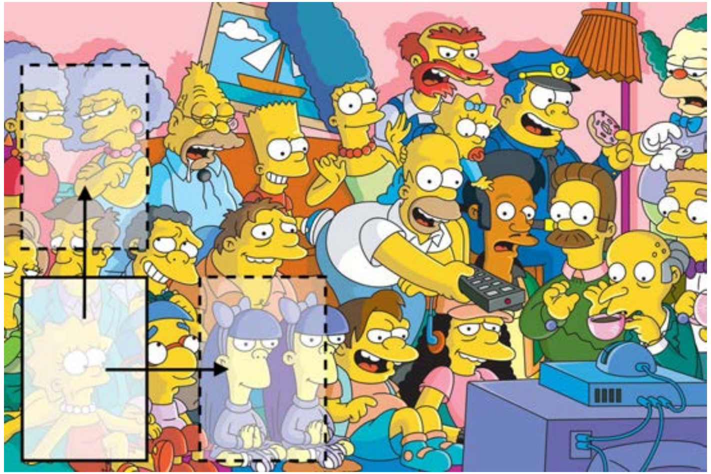
Source: https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568

Object Detection: Faster R-CNN

Use sliding window and apply a CNN to many different crops of the image?

Inflexible Size

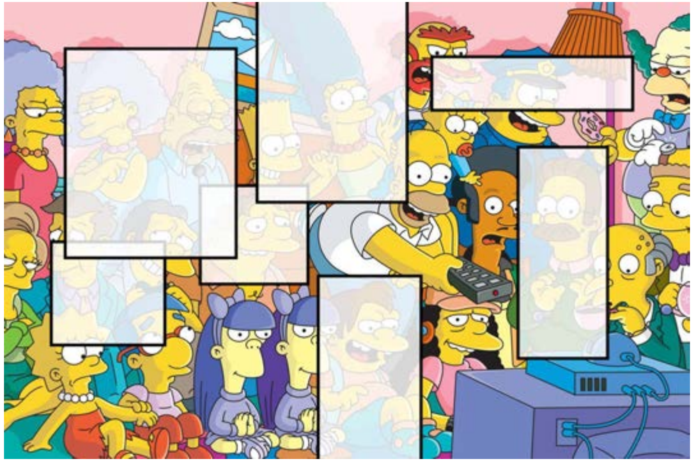
Too computationally expensive



Fixed Region Proposals?

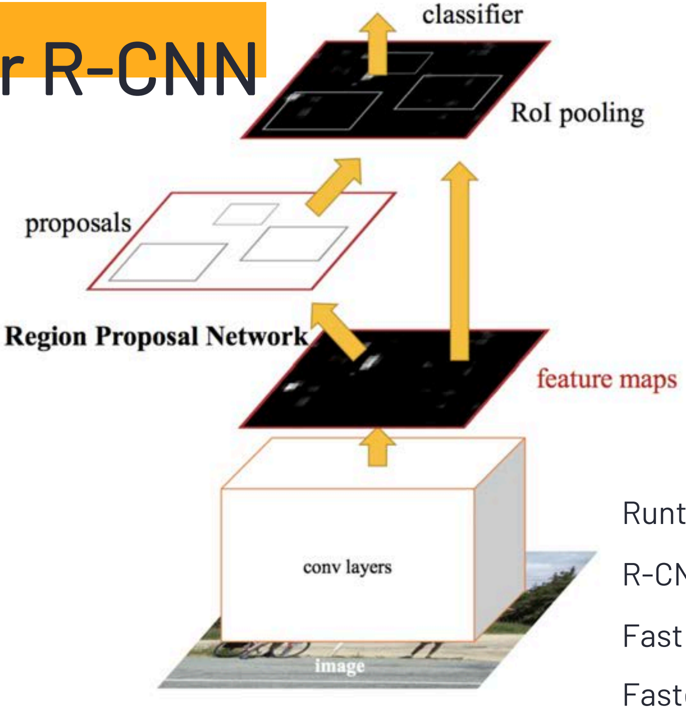
Select 2000 region proposals in a few seconds on CPU and apply a CNN to each one of them?

Better, but not enough! The region proposals should be learned



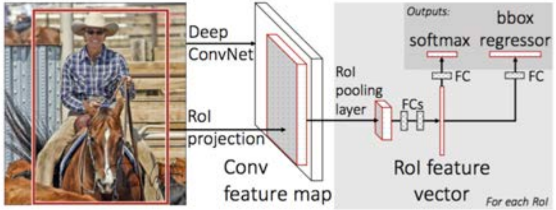
Object Detection: Faster R-CNN

Faster R-CNN



Faster R-CNN

Fast R-CNN



Fast R-CNN

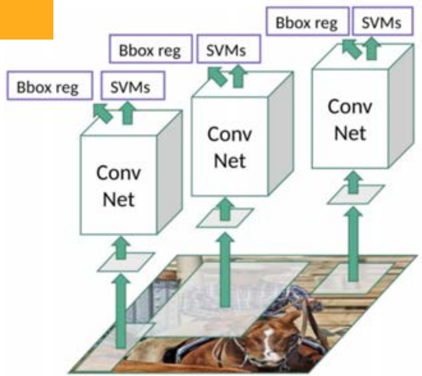
R-CNN

Runtime of the same task:

R-CNN: 49 hours

Fast R-CNN: 2.3 hours

Faster R-CNN: 0.2 hours



R-CNN

Part I – Simpson Characters Classification

Evaluation Metrics

- Test Accuracy
- Test Loss
- Precision
- Recall

VGG 16

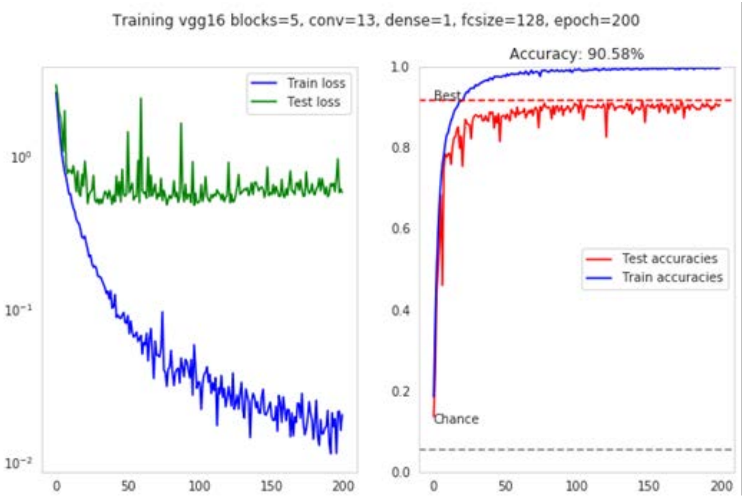


Image size (32,32)

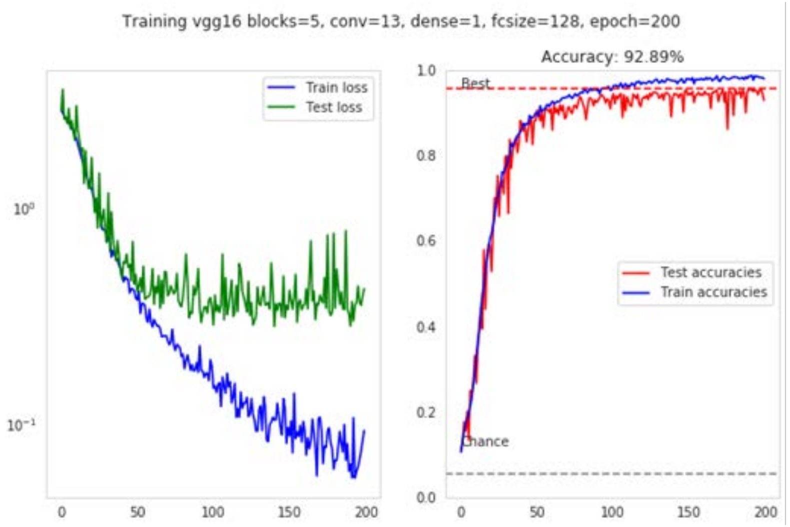


Image size (128,128)

VGG 19

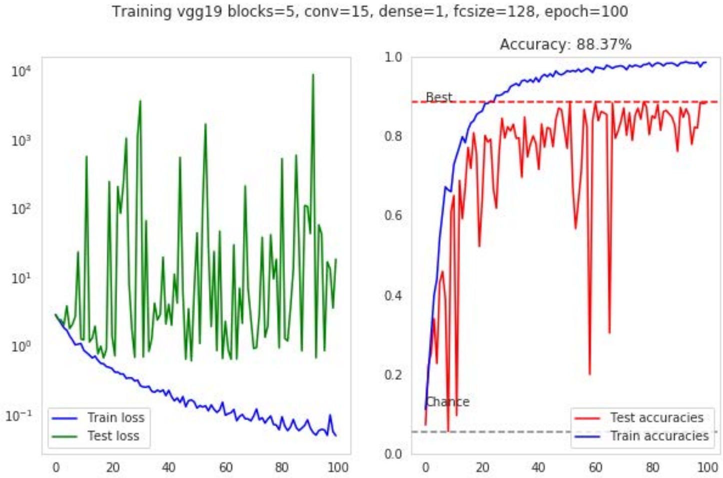


Image size (32,32)

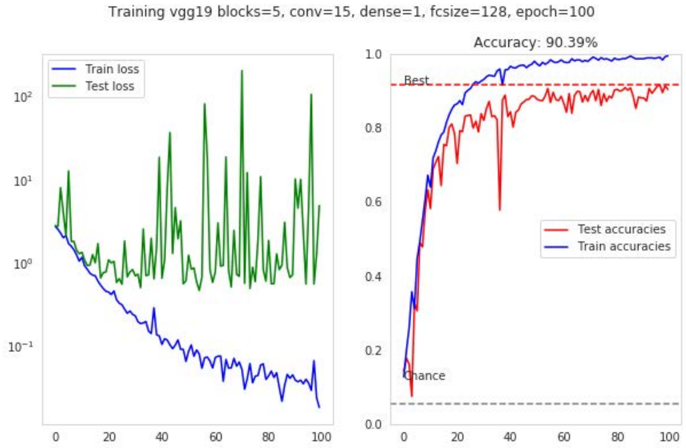


Image size (128,128)

Xception: Experiments

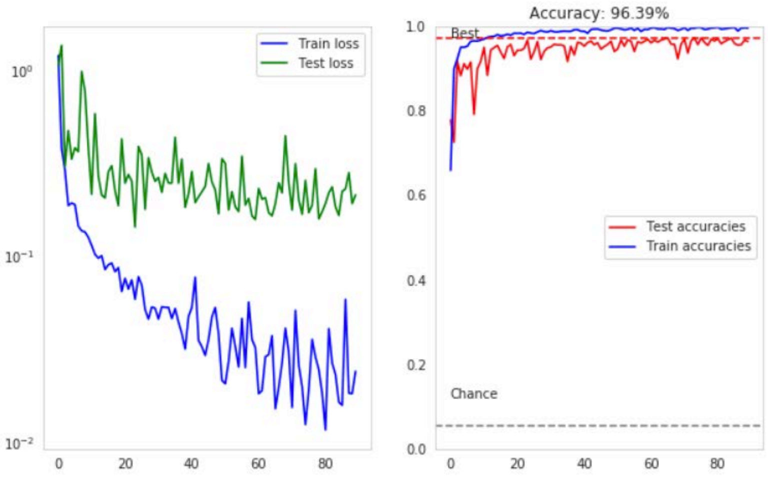


Image size (128,128)
No augmentation

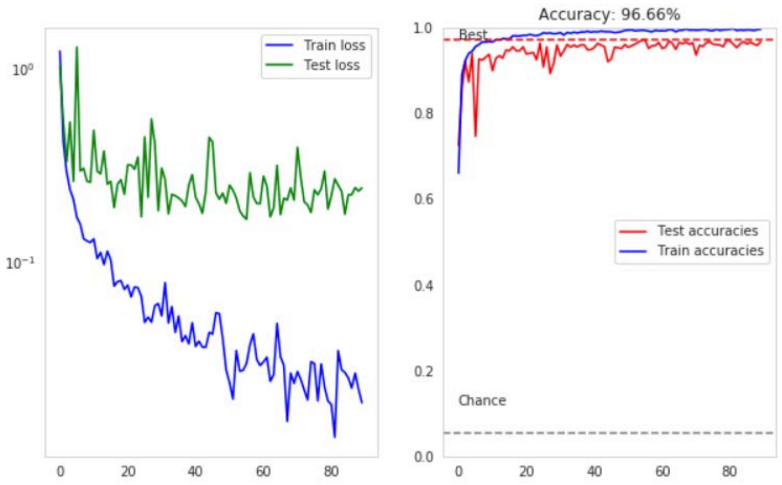


Image size (256,256)
With augmentation

Best Model: Loss Curve

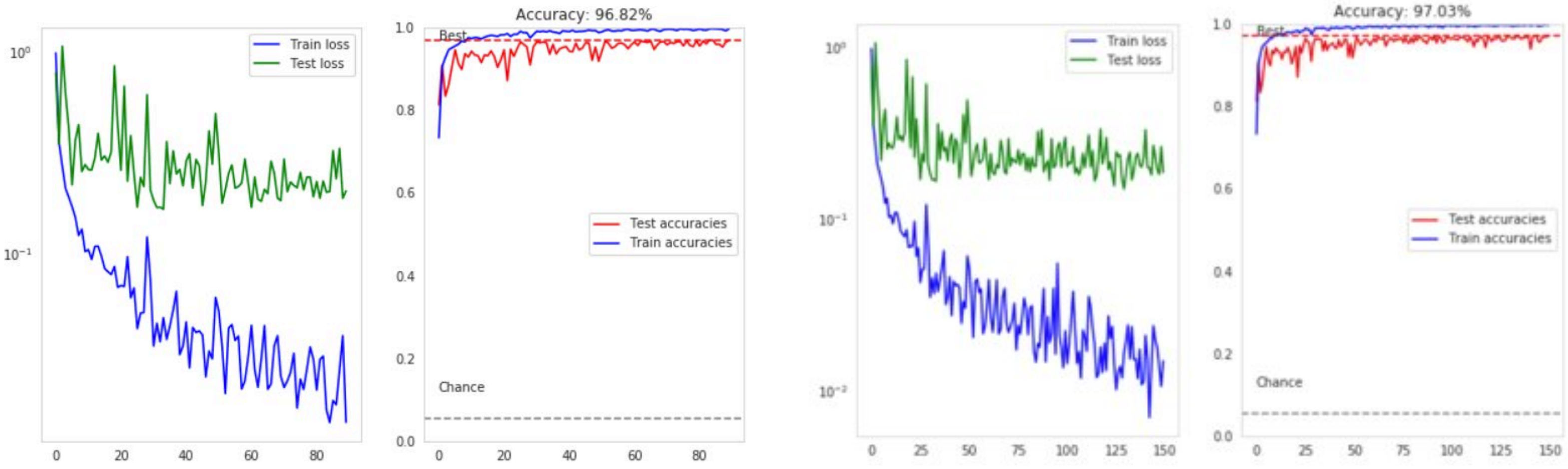


Image size: (128,128)

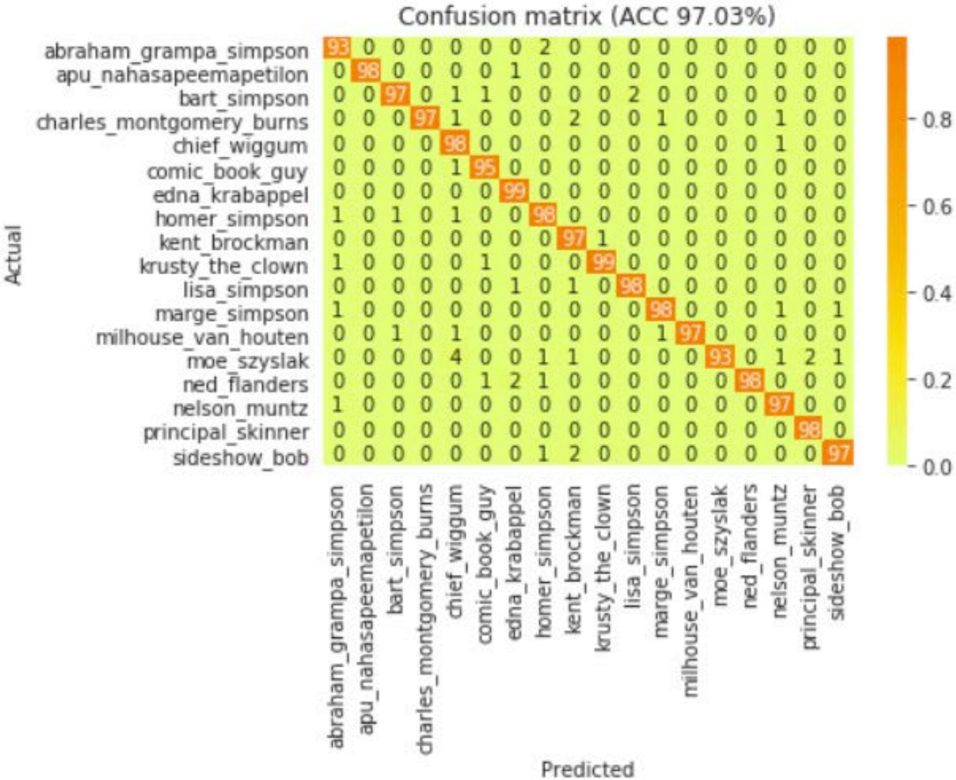
Augmentation:

randomly shift images horizontally and vertically

flip images horizontally

Test loss curve stabilized after ~50 epochs and test accuracy achieved 97.03% after 150 epochs

Best Model: Confusion Matrix



Performed well in general

Relatively poorly on:

Grampa Simpson (often confused with Homer Simpson)

Best Model: Heatmap

Correct Classification

Misclassification

Most of the time, the model focusing on the correct part of the image

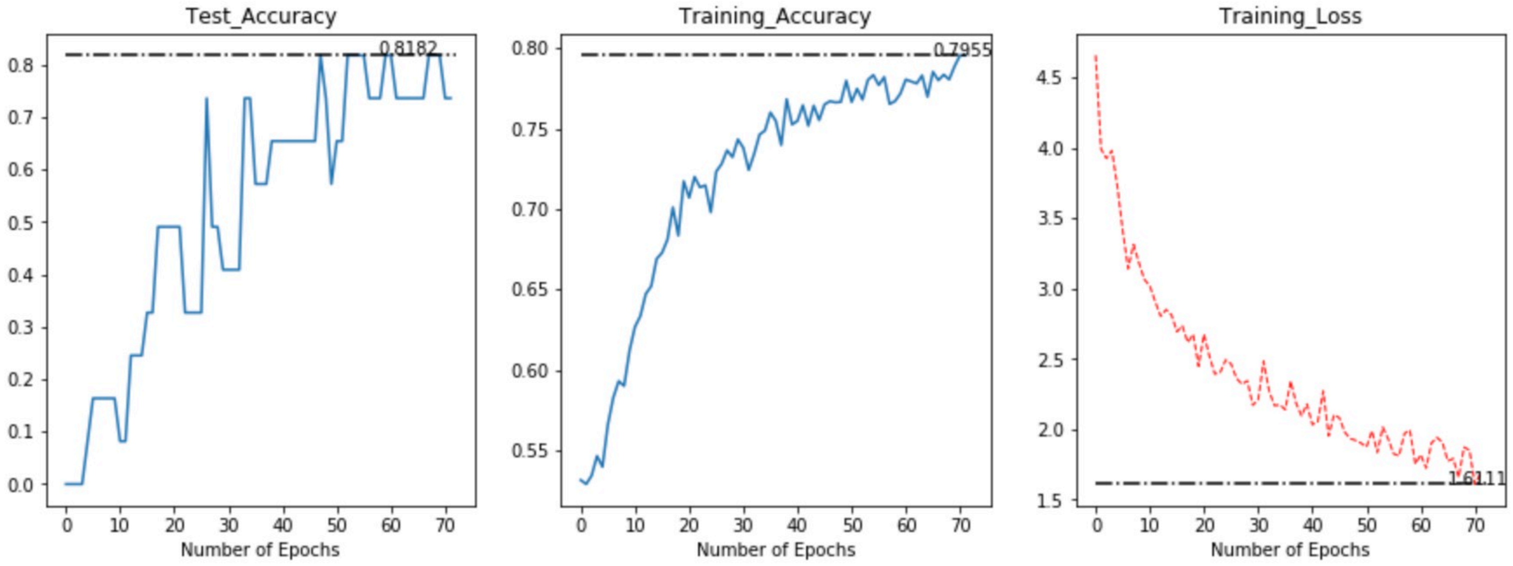
The model was unable to classify correctly when dissecting background

Part II - Object Detection and Classification

Evaluation metrics

- Model level
  - Accuracy score = number of accurately classified characters / actual number of bounding boxes
- Character level
  - Precision: fraction of relevant characters among the retrieved instances
    - true positive cases / predicted positive cases
  - Recall: fraction of total amount of relevant characters that were actually retrieved
    - true positive cases / actual positive cases
  - F1-score: a special weighted average of precision and recall

Best Model: Fast R-CNN



Part II Output Summary

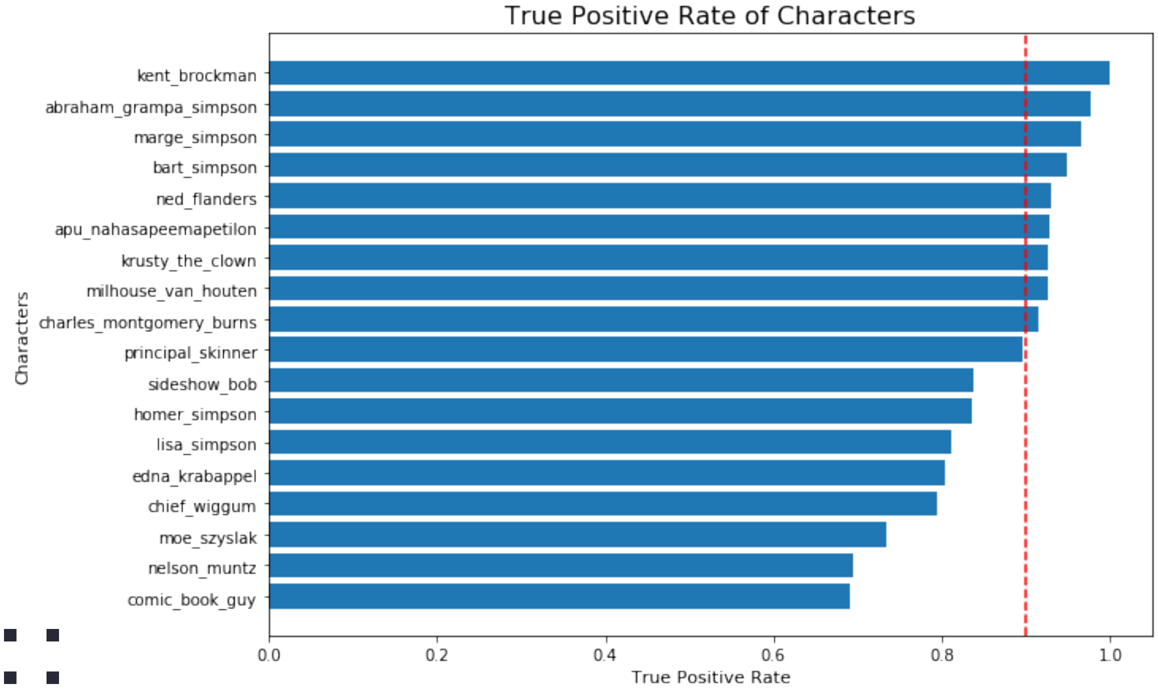
Test Metrics Performance - Top Six Accurate Characters			
Character	Precision	Recall	F1 score
ned_flanders	0.913	0.929	0.921
marge_simpson	0.832	0.966	0.894
kent_brockman	0.800	1.000	0.889
principal_skinner	0.870	0.895	0.883
krusty_the_clown	0.826	0.927	0.874
chief_wiggum	0.939	0.795	0.861

Part II Output Summary

Test Metrics Performance - Bottom Six Accurate Characters

Character	Precision	Recall	F1 score
abraham_grampa_simpson	0.636	0.977	0.771
charles_montgomery_burns	0.637	0.914	0.751
nelson_muntz	0.806	0.694	0.746
bart_simpson	0.541	0.949	0.689
moe_szyslak	0.629	0.733	0.677
comic_book_guy	0.547	0.690	0.611

Part II Output Summary

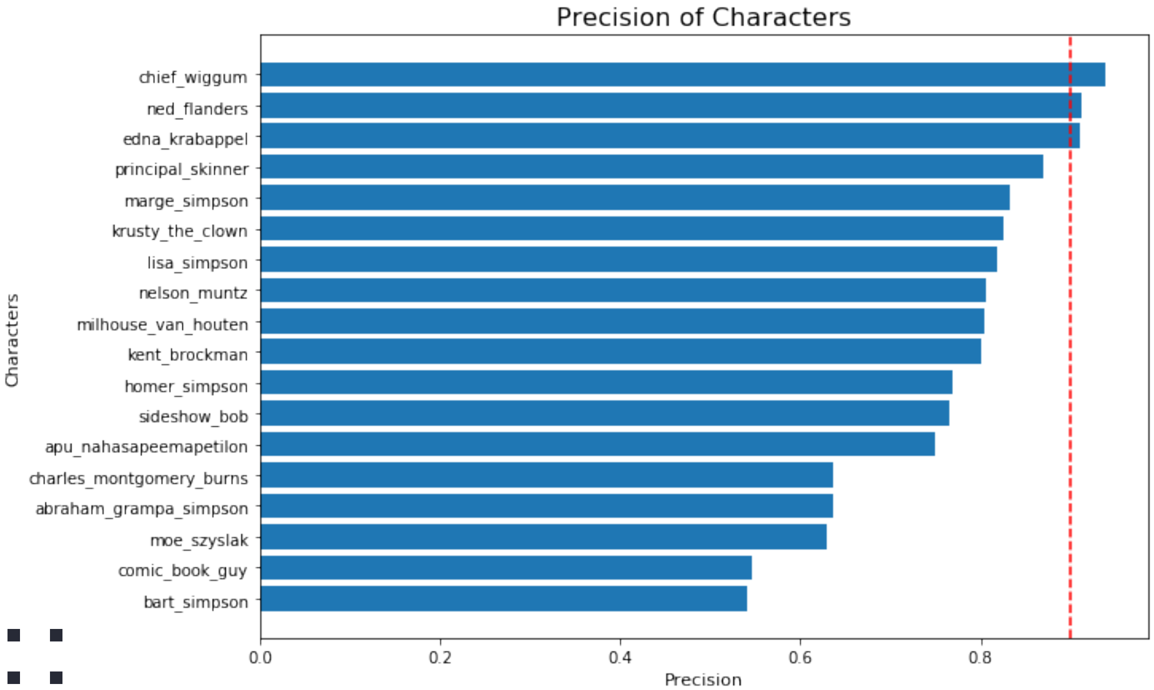


Half of the characters have recall greater than 90%.

The characters from Simpson Family (Marge, Bart, etc.) are highly likely to be identified.

The ranks of Simpson characters do not show significant difference.

Part II Output Summary

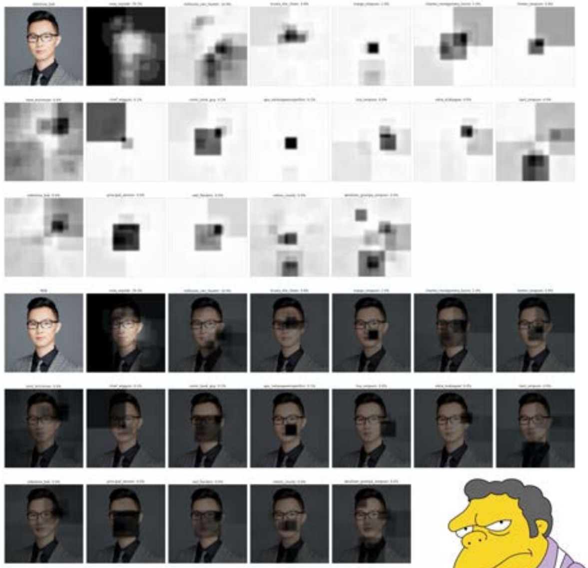


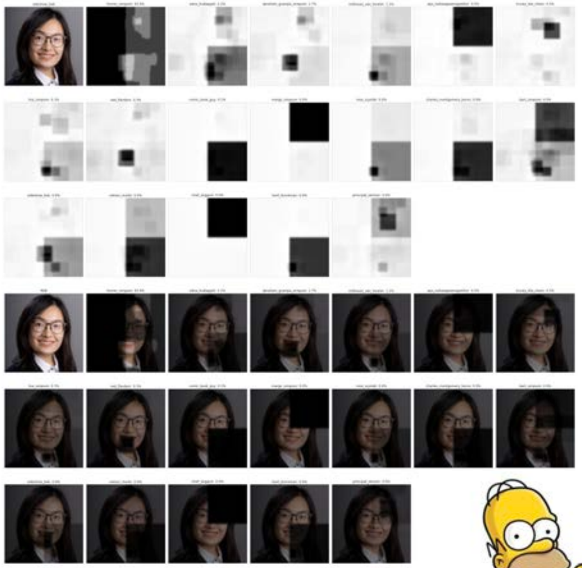
Only three characters have precision greater than 90%.

Some major characters (Bart Simpson) only have precision below 60%.

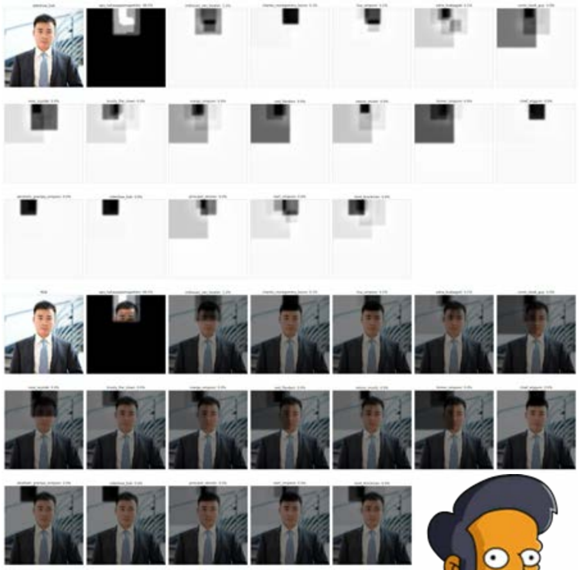
The model tends to predict more boxes as the major characters such as Bart Simpson and Abraham Grampa Simpson.

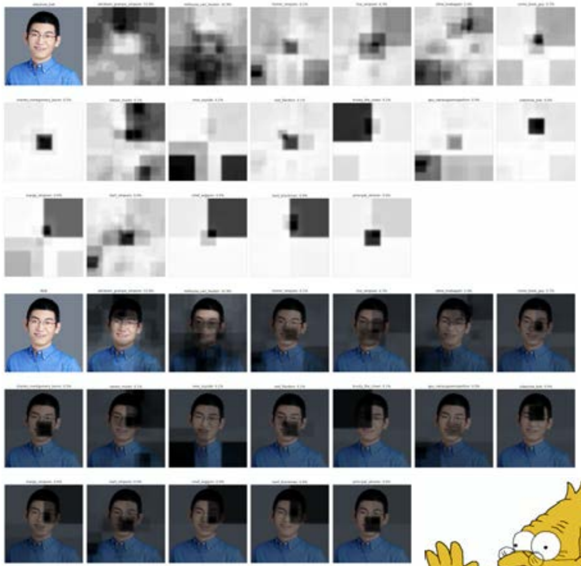
Easter Egg



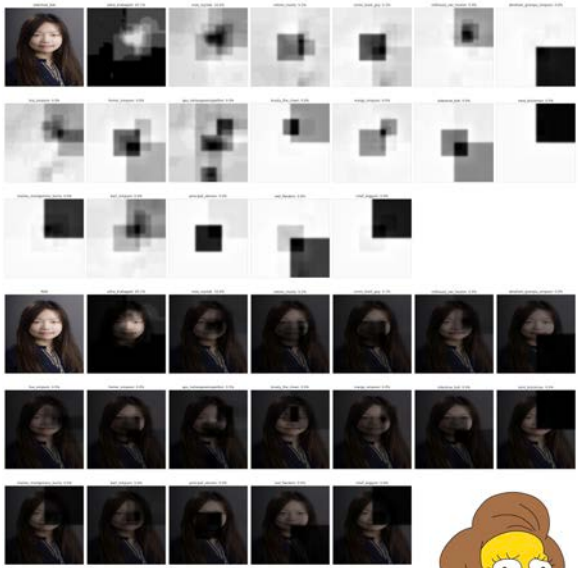


Easter Egg





Easter Egg



Conclusion - Classification

Great performance for classification (Part I)

Misclassification due to multiple characters in the same image

CNN algorithms play an important role.

Conclusion - Detection

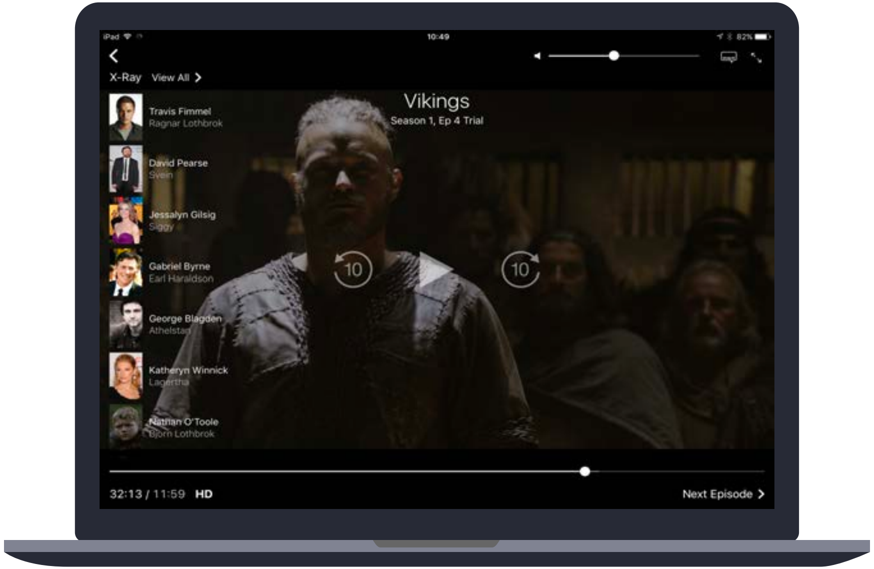
Good performance for detection (Part II)

Errors due to missing bounding box labels in the pictures.

Training and predicting takes incredibly long

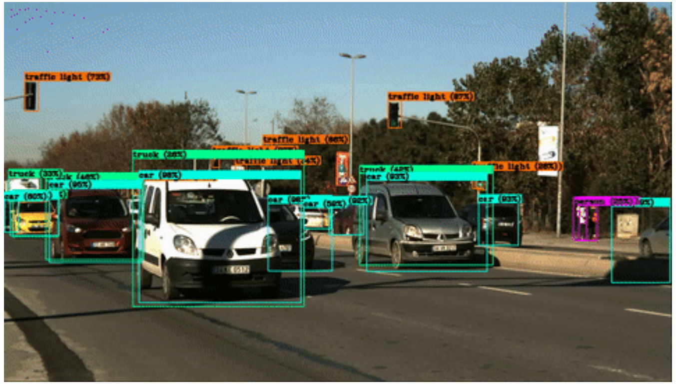
Application

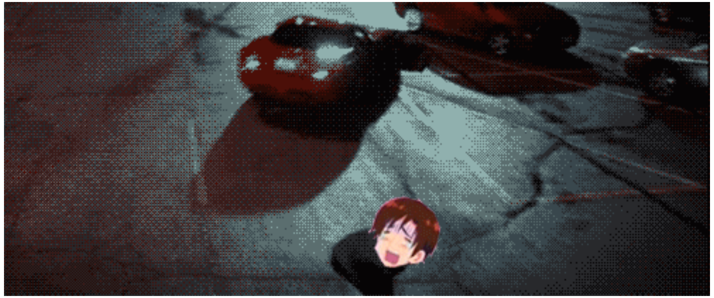
Real time TV show character detection for people not good at remembering names



Future Opportunities

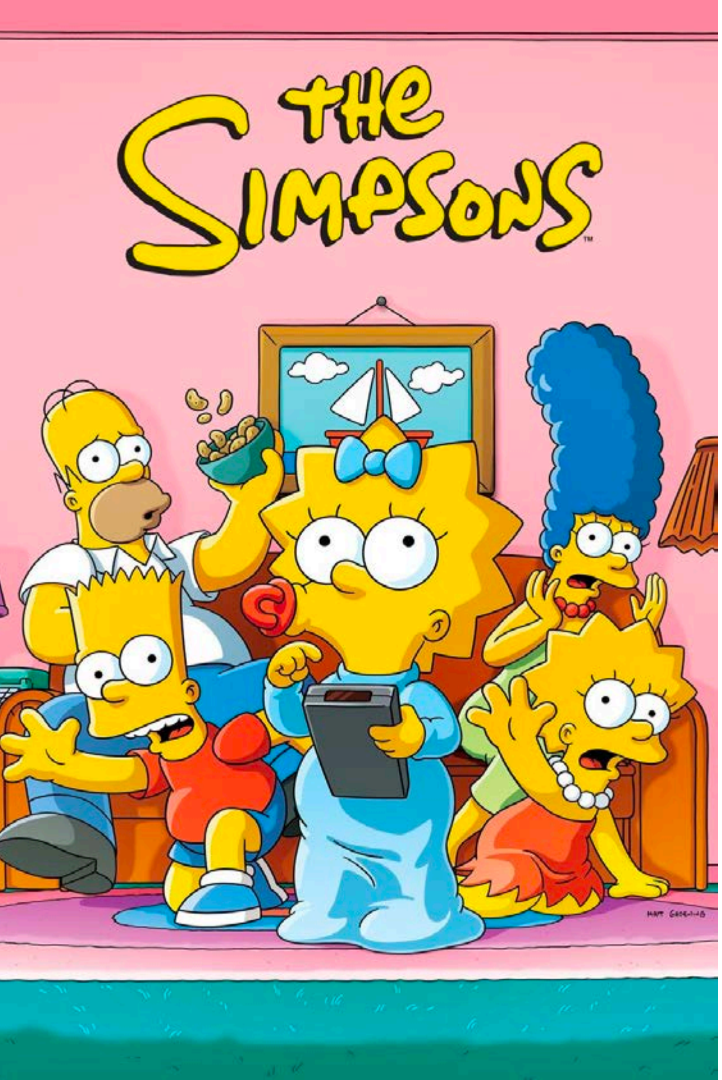
Faster Algorithm: YOLO (v4)





References

- Attia, A. (2017, June 12). The Simpsons characters recognition and detection using Keras (Part 1). Retrieved from https://medium.com/alex-attia-blog/the-simpsons-character-recognition-using-keras-d8e1796eae36
- Carremans, B (2018, August 17). Classify butterfly images with deep learning in Keras. Retrieved from https://towardsdatascience.com/classify-butterfly-images-with-deep-learning-in-keras-b3101fe0f98
- He, K et al. (2015). Deep Residual Learning for Image Recognition. Retrieved from https://arxiv.org/pdf/1512.03385.pdf
- Krizhevsky, A et al. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
- Simonyan, K and Zisserman, A (2015).Very Deep Convolutional Networks for Large-Scale Image recognition. Retrieved from https://arxiv.org/pdf/1409.1556.pdf
- Szegedy, C et al. (2014). Going deeper with convolutions. Retrieved from https://arxiv.org/abs/1409.4842



Thanks!

Any questions?