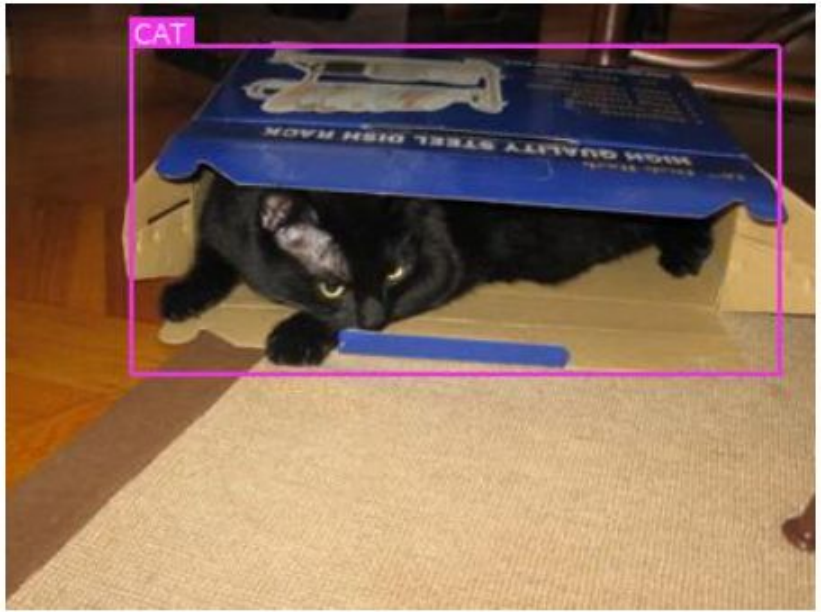
Large Scale Object Detection

Èlia Armengol, Gabriela Cid, Jordi Marín

Introduction

Our objectives are:

- Implement a code that is able to detect and localize objects.
- Improve the accuracy without altering the efficiency.

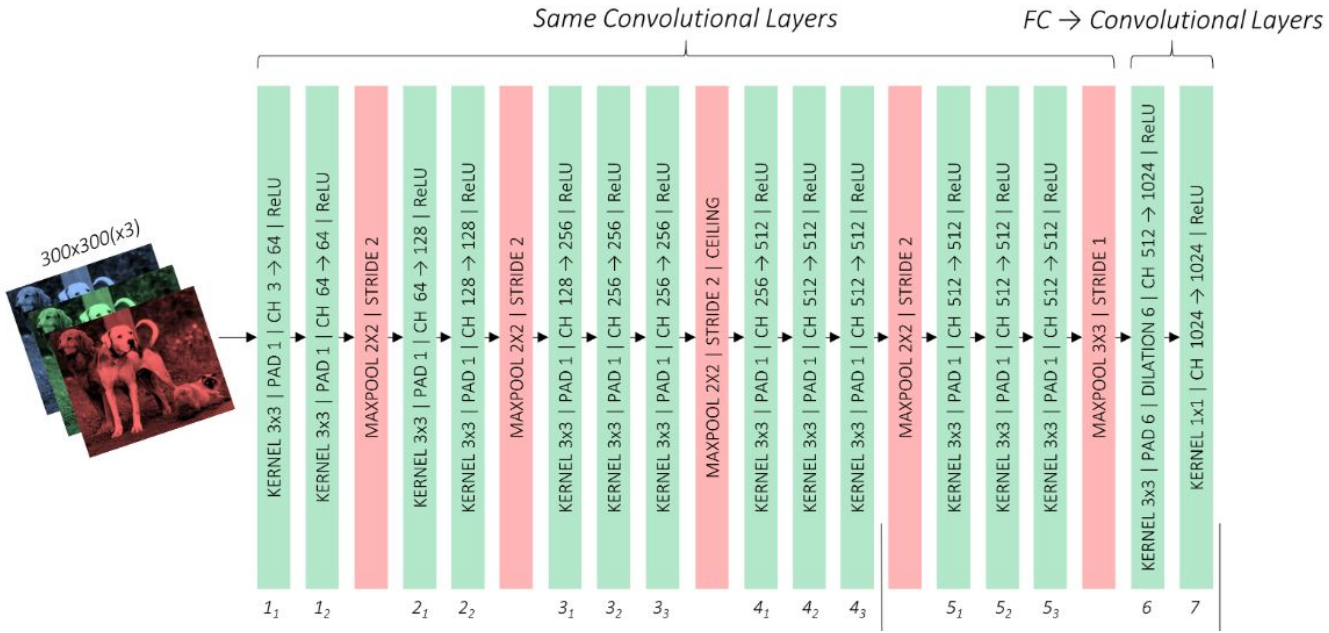


Detected objects

- Aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, potted plant, sheep, sofa, train and tv monitor

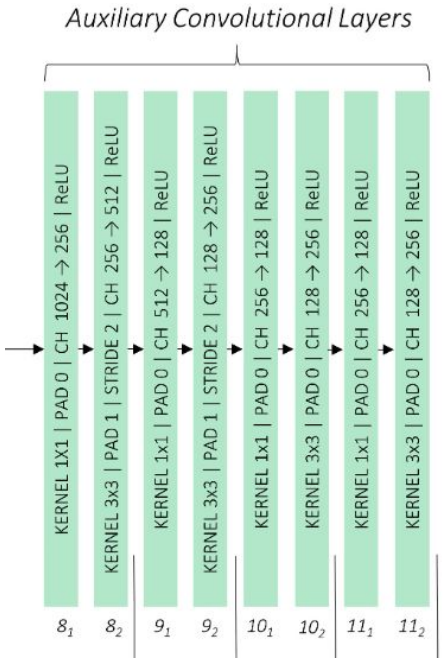
Implementation

We have a model composed of: **Base convolutions**.



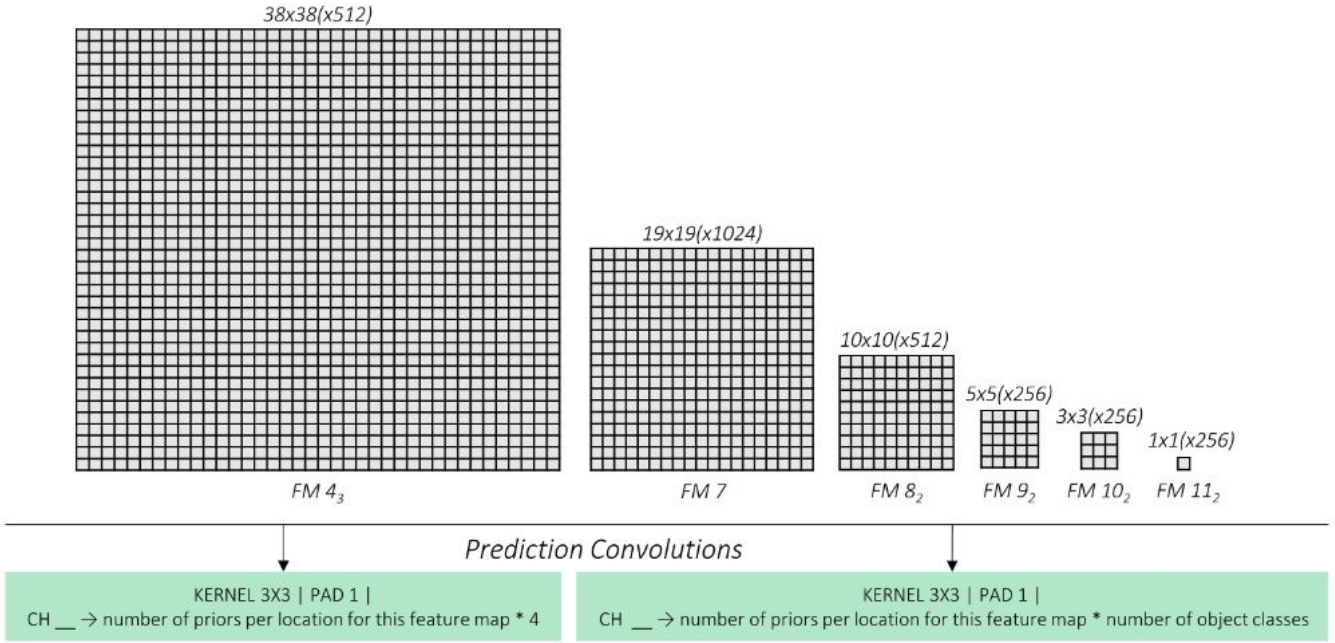
Implementation

We have a model composed of: **Auxiliary convolutions**.



Implementation

We have a model composed of: **Prediction convolutions**.

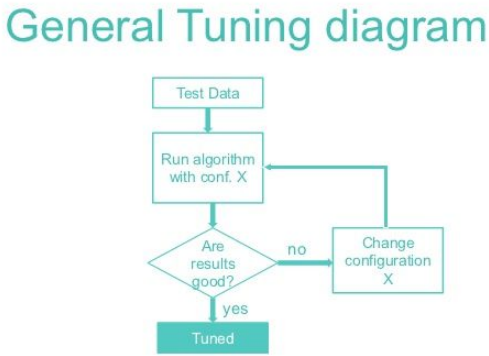


Changes

How can the accuracy be improved?

- Data

- Algorithm tuning / Play with architecture



Changes

Data approach**:**

- Data transformation

    Random Zoom chance

    Random Horizontal flip chance

- ~~More data~~
- ~~Normalize~~

Changes

Algorithm tuning approach:

- Number of epochs

- Learning rate parameter (Initial and decay)

- Momentum

Changes

Architectural approach:

- Base convolutions

- Auxiliary convolutions

    - Layer size reduction speed

- Prediction convolutions

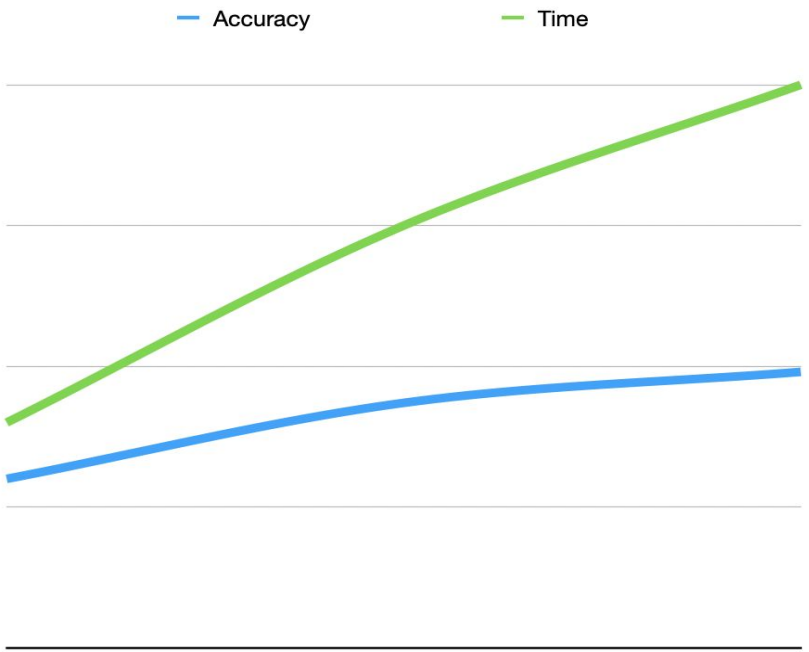
Conclusions

We tried different aspects of the code to better our performance and this is what we saw:

Conclusions

Bigger Epoch

➜ More accuracy but the time it took to run the program was very inefficient and we think is because we have a very large dataset.



Conclusions

Transformation

➜ Zoom

➜ Horizontal Flip

Layers

➜ Changing auxiliary convolution's dimensions.

Conclusions

Transformation

➜ Zoom

➜ Horizontal Flip

Layers

➜ Changing auxiliary convolution's dimensions.

Accuracy does not change

Conclusions

As we have shown during the tests it took a lot of time only for a bit of improvement in the accuracy.

Therefore we conclude that the first implementation was the best one.

Thank you for your attention

Any questions?