# IACV Project

## sea lion classification 2017/2018


Instructor: Vincenzo Caglioti

Supervisor: Giacomo Boracchi
Diego Carrera

Team: Zhou Yinan 10551242

Wang Xuan 10549075

Sofia Mitoulaki 10597553

# Introduction

- This project is based on a Kaggle competition. The original competition requires competitors to develop an algorithm which accurately count the number of sea lions in aerial photographs.
- Our project can be seen as the first step of the counting algorithm, which takes a patch extracted from the image and classifies it as certain sea lion type or background.
- There are totally 5 sea lion types: adult males, subadult males, adult females, juveniules, pups

# Images

in this image, sea lion counts

- adult males = 17
- subadult males = 6
- adult females = 251
- juveniles = 17
- pups = 225

- totally 947 images
- Each image of size 4K * 3K
- Each image ~ 10 MB

# Background Knowledge

- In recent years, deep learning models for image classification is a hot topic.
- Since Alex net, deep CNN architectures improve the classification performance a lot. Now CNN is the state of art for image classification problem.
- Alex net, VGG, GoogleLeNet, ResNet are some popular deep CNN architectures for image classification.
- In our project, we develop our own CNN architecture for 6 class image classification.(sea lion + background). We also compare our model with VGG16

# Challenge

Each image has high resolution, put into CNN directly requires large computation power and RAM

solution: use slide window to extract patches and use patches as CNN input

Data not enough for deep CNN architecture

solution: data augmentation

Dataset is highly unbalanced

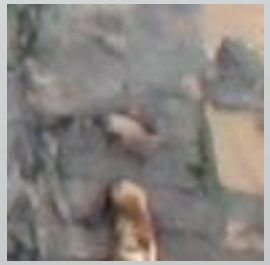
solution: down/up sampling & different class weights

# Implementation Details

# Data

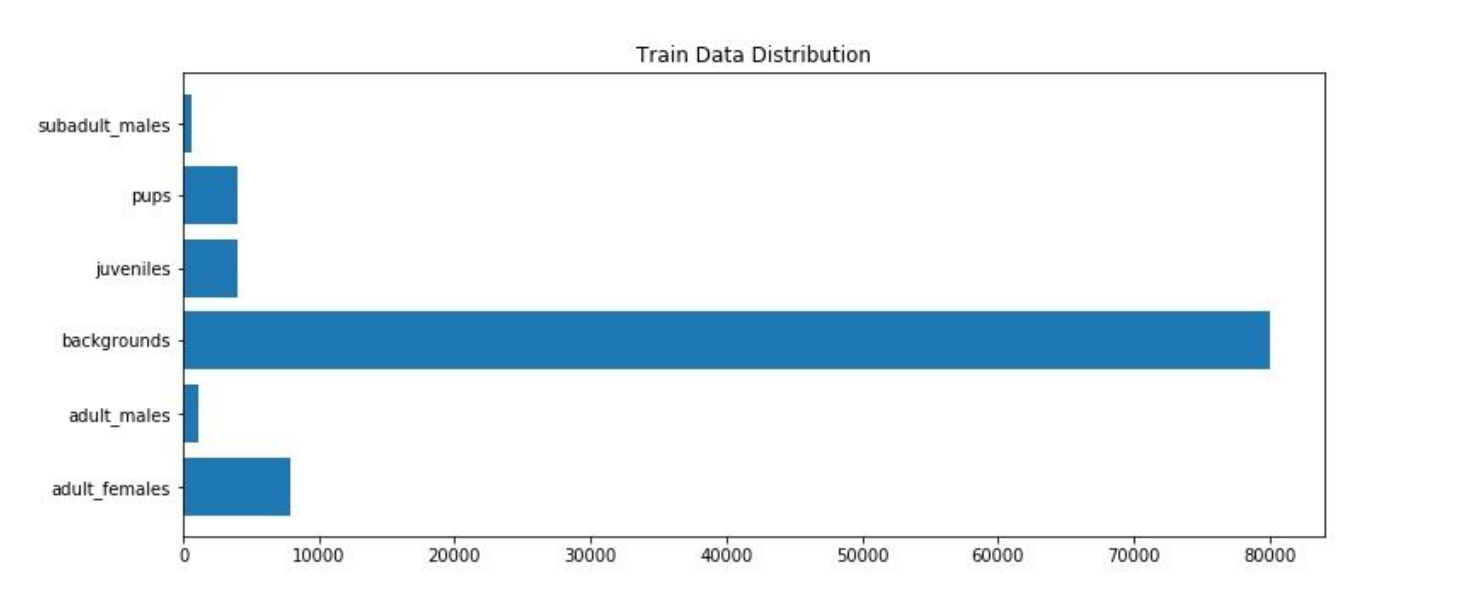
- Different sea lion with different colour label
- Use blob detection to get dot colour and coordinates
- Extract patches with sea lion at centre. Some examples:

adult male



pups

# Data




number of patches

- train : 172748
- valid: 42521
- test: 97666

valid, test data have similar data distribution

# Our model

| Layer | Operation | Input size | Filter size | Padding | Stride | Output size |
|---|---|---|---|---|---|---|
| Layer 1 | Convolution | 96*96*3 | 5*5*8 | 0 | 1 | 92*92*8 |
| | MaxPooling | 92*92*8 | 2*2 | 0 | 2 | 46*46*8 |
| Layer 2 | Convolution | 46*46*8 | 3*3*5 | 0 | 1 | 44*44*5 |
| | MaxPooling | 44*44*5 | 2*2 | 0 | 2 | 22*22*5 |
| Layer 3 | Convolution | 22*22*5 | 3*3*5 | 0 | 1 | 20*20*5 |
| | MaxPooling | 20*20*5 | 2*2 | 0 | 2 | 10*10*5 |
| Layer 4 | Convolution | 10*10*5 | 3*3*10 | 0 | 1 | 8*8*10 |
| | MaxPooling | 8*8*10 | 2*2 | 0 | 2 | 4*4*10 |
| Layer 5 | Fully Connected | 160 | / | / | / | 256 |
| | Fully Connected | 256 | / | / | / | 6 |

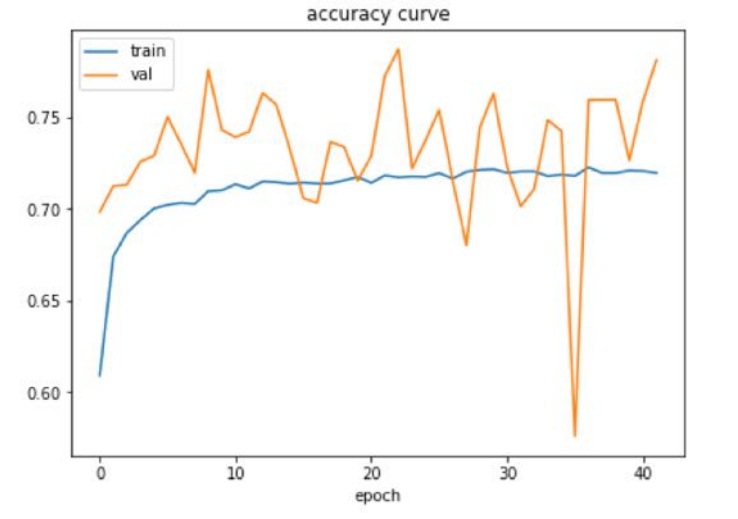
- After each conv, reLu activation is used
- Total parameters: 44421
- A softmax is added to the last for classification

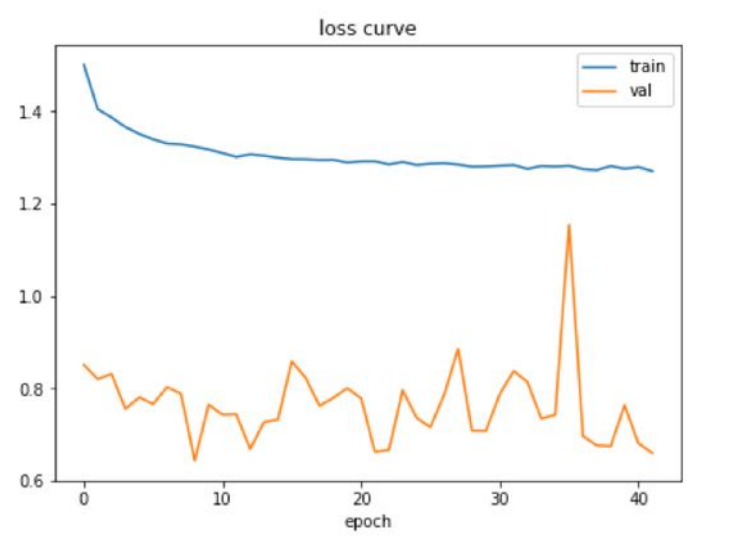
# Training Detail

- We use python, keras framework with tensorflow as backend
- Train 40 epochs
- Optimizer: Adam
- Learning rate: 1e-4

# Result Analysis

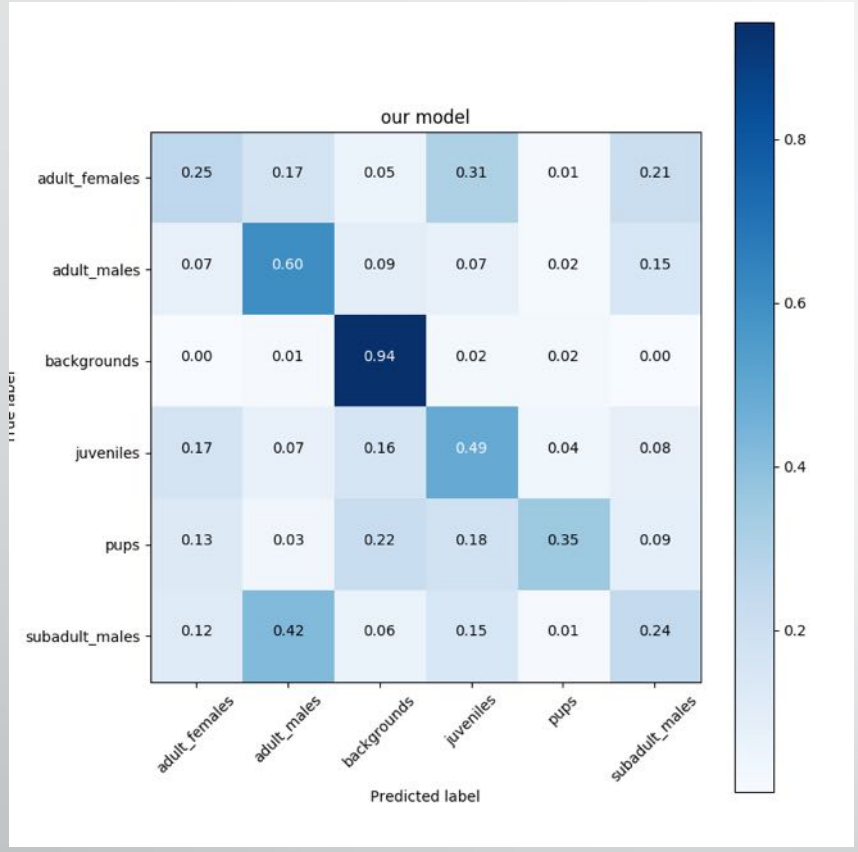
# Train Curve




- valid score fluctuates
- Try larger batch size and smaller learning rate

# Confusion Matrix

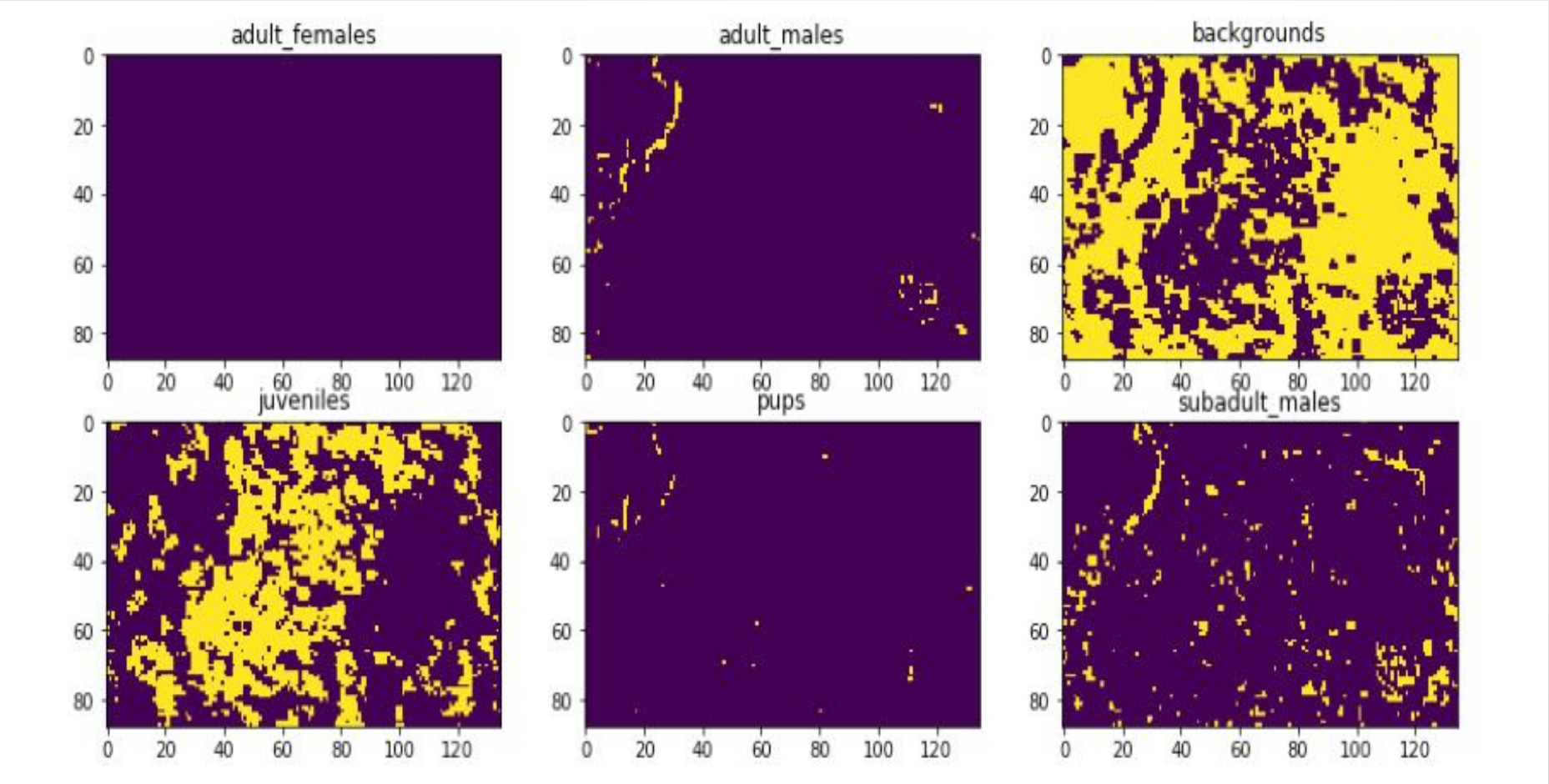



- background: easier to detect
- subadult males confused with adult males, probably because of the similar size
- most sea lion classes are classified correctly

# Semantic Segmentation

Use our model in fully convolutional way to generate 6 heatmaps for this image

The result is shown in the next page




we can see that our model classifies the backgrounds quite well, the shape of the island is extracted. But for sea lions in this image, they are not classified so well.

# Error Analysis

- Pups confused with backgrounds: they are small and look like rock
- adult females confused with juveniles: they share similar sizes
- subadult males confused with adult males: similar sizes

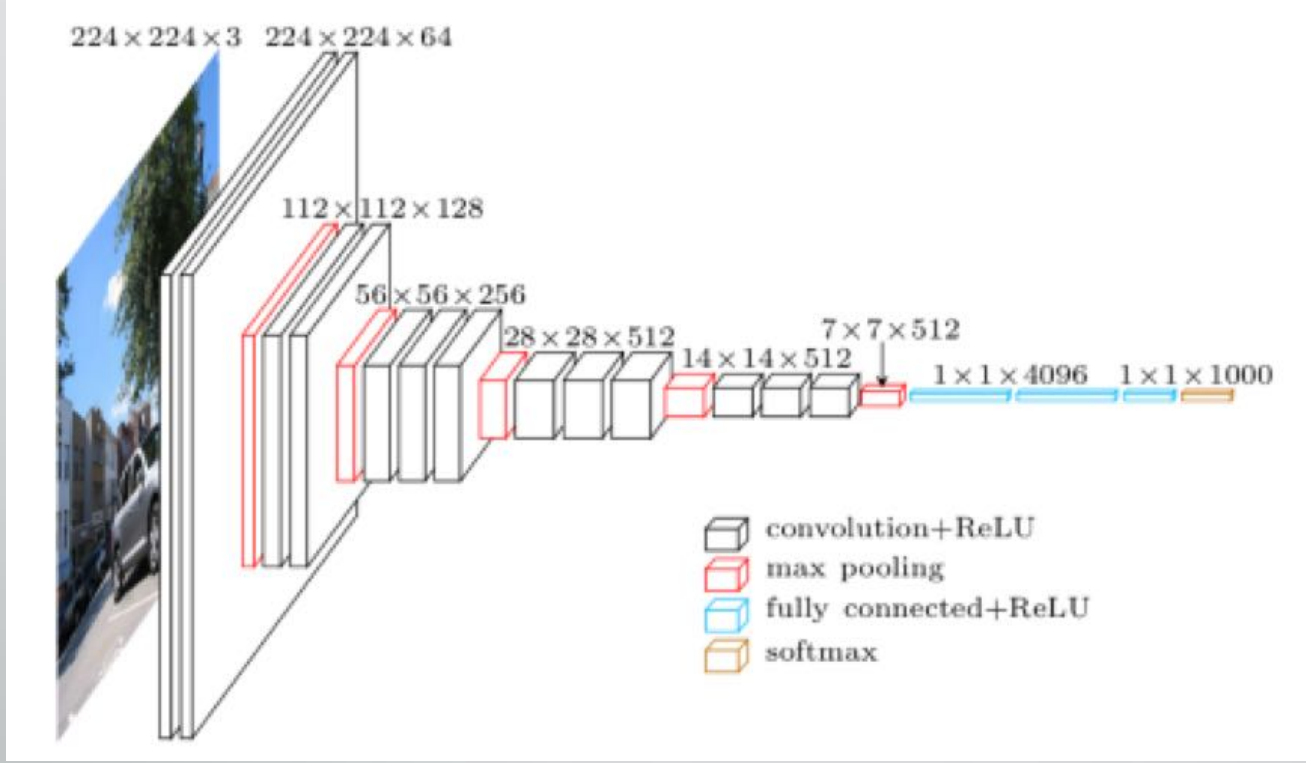
It seems that size is an important characteristic to classify sea lions. In order to deal with the above issues, we tried data augmentation with some more particular sea lion class samples, but there is no significant performance improvement. So we tried with more complex model and transfer learning using VGG16.

# Transfer Learning

- Transfer learning is quite popular in compute vision deep CNN model. Because CNN model can extract abstract features from images. We can use pre-trained model in different tasks. With fine tuning, we don’t need to train a model from scratch.

- In our experiment, we tried with VGG16 model.

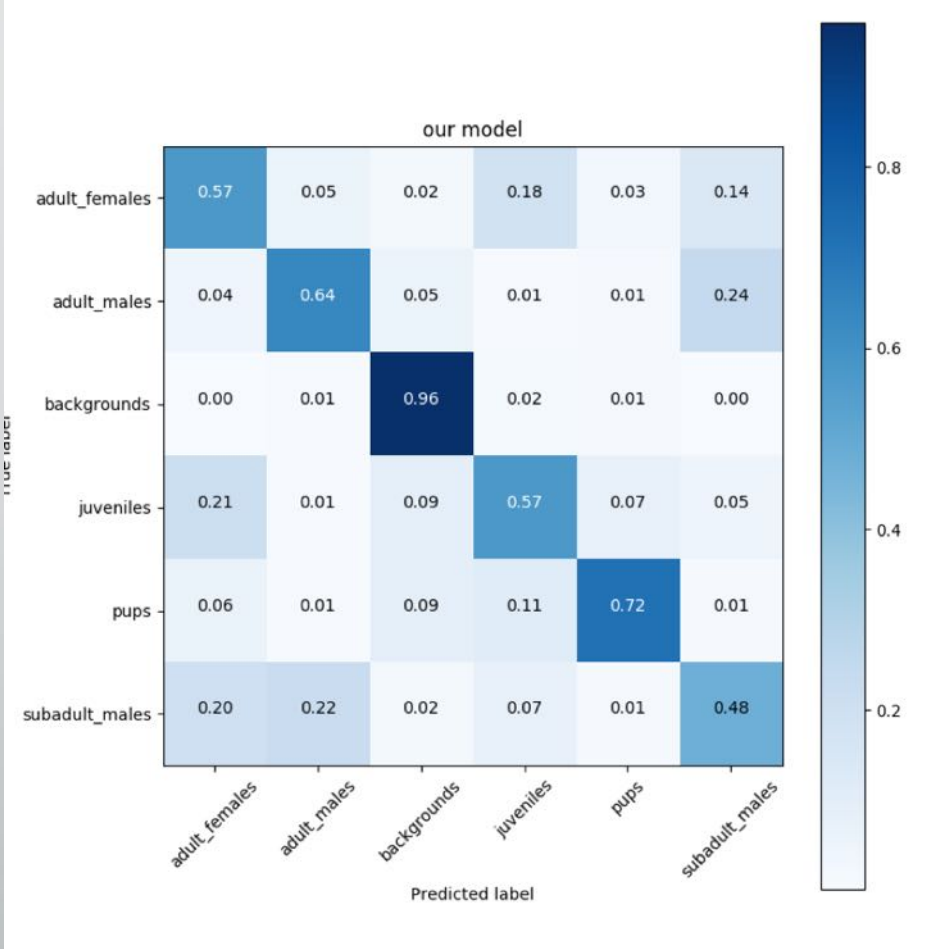
# VGG 16




- vgg16 is much more deeper than our model
- We discard the fully connected layer and add our own

# Confusion Matrix

## VGG16




We first freeze all the convolutional layers and train only the fully connected layers. Then we decrease the learning rate to train the whole network.

As we can see, vgg16 model achieves better score than our model.

# Conclusions

In this project, we build a CNN architecture from scratch using keras framework. We face a 6-class classification problem and the dataset is not balanced. We use sampling and weighting techniques to improve the performance. We experimented with different optimization algorithms and finally chose Adam as our final approach. Finally we tried transfer learning with vgg16.

# Issues and Further Work

- Counting problem: we treat the problem as a classification problem but in each patch sea lions can overlap. In order to deal with is issue, we can use overlapping slide-window or regression.
- Spatial Correlation: pups seem to show up with its mom. Adult males seem to be alone. We can use this prior knowledge to improve the performance.
- Object Detection: we can use object detection approach to treat this problem.

# Thank you

https://github.com/marioZYN/IACV-Project