Machine Learning Engineer Nanodegree

Capstone Project Report

Kamil Kaczmarczyk - September 10th, 2017

Vision-Perception Module

Situational Awareness System for Autonomous Vehicles

I. Definition

Project Overview

Goal and preface:

Goal of this project to provide vison basec perception module of a situatioanl awareness for Autonomous Vehicles or in other words to detect few key classes of objects that could potentially be obstacles for an undisturbed and safe flight of an autonomous aircraft or a drone. The approach applied relies on computer vision and machine learning methodologies. Situational Awareness System for Autonomous Vehicles could be thought of also in terms of famous sense-and-avoid systems for drones. Their goal is to detect obstacles and avoid them in real time during flight to allow for safe drone operation and for successfull mission. In this project the goal however is not to avoid obstacles but simply to detect them and define their position within a frame of a video stream.

Background information and key motivation:

Sense and avoid systems have been pursuied for a long time in industry and they are one of the key technologies missing to enable fully autonomous operations of drones. In order to perform fully autonomous operations in the air it is necessary to have a situational or environmental awareness around the vehicle. This is relevant to know where the "flyiable" space is around the aircraft so that safe operations without crushing into obstacles or terrain is possible but also it requires to understand where mission relevant objects or places are in the vehicle environment such as a potential landing spot or an object to be followed and tracked. Many approaches have been adopted to detect obstacles on the flight path and most of them directly relied on sensing of the environment and some signal processing using active radar, laser, optical sensor, sonar or passive radars. Recently more and more demos from the academics and industry relies on computer vision and camera approaches. Till this day however there is no high reliability sense and avoid system on the market, few companies are making good strides towards that but this key technology is still missing today. More information and sense and avoid market solutions can be found in the article: http://droneanalyst.com/2016/09/22/sense-and-avoid-for-drones-is-no-easy-feat/

How the problem could and should be solved:

Recent progress in computer vision approaches based on machine learning could perhaps enable to solve the problem of sense and avoid and in particular of the situational awareness. This work is an attempt to detect a small subset of objects namely aeroplanes and birds in a frame of a video stream from a flying aircraft. If however this problem is to be fully solved then this approach should be further extended into other objects; into sensor fusion (perhaps with radar, or other sensors); maps could be added as well; small transponder devices and air traffic management solution should be studied or finally Vehicle to Vehicle communication could be also considered. This project is just the first building brick for detecting a subset of objects using CV.

Academic research:

Sense and avoid aircraft detection using computer vision:

- Vision4UAV Research Group from Technical University in Madrid (http://138.100.76.11/visionguided2/?q=node/349) have produced work with sense and avoid using computer vision. Their dataset of few movies is used for demonstration of the results in this work:

- [1] "A Ground-Truth Video Dataset for the Development and Evaluation of Vision-based Sense-and-Avoid Systems"; Adrian Carrio, Changhong Fu, Jesus Pestana, Pascal Campoy; 2014 Technical University in Madrid

- [2] "SIGS: Synthetic Imagery Generating Software for the Development and Evaluation of Vision-based Sense-And-Avoid Systems"; Adrian Carrio, Changhong Fu, Jean-Francois Collumeau, Pascal Campoy; 2015

- Carnegie Mellon University - overview of the sensor technologies and Sense and Avoid requirements from 2008-2009 as well as passive system for long range vision based detection of objects without the machine learning approach:

- [3] "Prototype Sense-and-Avoid System for UAVs"; Christopher Geyer, Debadeepta Dey, Sanjiv Singh; 2009 Carnegie Mellon University

- [4] "Avoiding Collisions Between Aircraft: State of the Art and Requirements for UAVs operating in Civilian Airspace"; Christopher Geyer, Sanjiv Singh, Lyle Chamberlain; 2008 Carnegie Mellon University

- [5] "Passive, long-range detection of Aircraft: Towards a field deployable Sense and Avoid"; Debadeepta Dey, Christopher Geyer, Sanjiv Singh, Matt Digioia; 2009 Carnegie Mellon University

System

- Korea Aerospace Research Institute and KAIST - Department of Aerospace Engineering research on vision based sense and avoid using computer vision but without machine learning approaches but particle filters instead:

- [6] "Vision-Based Sense-and-Avoid Framework for Unmanned Aerial Vehicles"; SUNGSIK HUH, SUNGWOOK CHO, YEONDEUK JUNG, DAVID HYUNCHUL SHIM; 2015 Korea Aerospace Research Institute and KAIST - Department of Aerospace Engineering

- Ecole Polytechnique Federale de Lausanne EPFL - sense and avoid using both appearance and motion cues but no classification performed using machine learning approaches:

- [7] "Detecting Flying Objects using a Single Moving Camera"; Artem Rozantsev, Vincent Lepetit, and Pascal Fua; 2015 Ecole Polytechnique Federale de Lausanne EPFL

Sense and avoid market overview:

- Overall market overview here: http://droneanalyst.com/2016/09/22/sense-and-avoid-for-drones-is-no-easy-feat/

- Iris Automation (http://www.irisonboard.com) - Startup backed by Y Combinator to solve the sense and avoid with the help of computer vision. For now however they have not produced a demo to show their technology.

Birds detection using computer vision and machine learning:

- The University of Tokyo study for birds detection near wind turbines with CNN networks but on a limited static dataset of up to 70 images and with a static perspective:

- "Evaluation of Bird Detection using Time-lapse Images around a Wind Farm"; Ryota Yoshihashi, Rei Kawakami, Makoto Iida, and Takeshi Naemura; The University of Tokyo

- "Detection of small birds in large images by combining a deep detector with semantic segmentation"; Akito Takeki, Tu Tuan Trinh, Ryota Yoshihashi, Rei Kawakami, Makoto Iida and Takeshi Naemura; The University of Tokyo

Problem Statement

Problem:

- detect and identify birds and aircraft in a frame of a video taken from a Point of View of another flying aircraft as a part of a sense and avoid system for autonomous vehicles.

- problem can be measured by detections indicators in four classes (aircraft, bird, sky or ground). This allows to measure the performance of models and reproduce the results

- goal is to create either bounding boxes or heatmaps around the first two classes of aircraft and birds

Metrics

Metrics selected:

- main metric is accuracy of classification of an image into one of the four classes for the test dataset (aircraft, bird, sky or ground). This metric serves for both the benchmark and the improved models as the main quantitative metric for comparing of the models.

- for additional demonstration purposes a sample of images of the internet is used for which a sliding window technique with different scales of windows is applied and each of them is classified. This is to create a object detection heatmaps and on this basis create bounding boxes. Metric here is a visual human comparison and inspection of the correct classification and a bounding box localization.

- finally demo video of an aircraft flying in front of another aircraft flight path is used with a qualitative metric being an accurate object detection and localization using again the entire pipeline with a sliding window technique and heatmaps being later on used for bounding boxes.

II. Analysis

Data Exploration

Dataset used for this project consists of pictures downloaded from the popular web search engine google in the section of images. It contains and is divided into four distinct classes of pictures:

- aircraft pictures of Boeing 737 and Cessna 172 models in flight - 400 images

- birds pictures also in flight mostly on the background of sky - 367 images

- sky images containing either a clear sky or clouded and ocluded sky images - 407 images

- ground images containing a mix of various pictures from flight of cities, fields, mountains and other landscapes where most of the image area is covered by ground so that it does not contain a lot of sky in it - 407 images

Examples of each class can be seen here:

Aircraft

Sky

Bird

Ground

Step 1 - Aspect Ratios Investigations & Conversion

For each class a simple check was performed to inspect the basic characteristics of aspect ratios of images as this information is going to be relevant for the future steps of processing.

	aircraft images aspect ratio information	birds' images aspect ratio information	sky images aspect ratio information	ground images aspect ratio information
Max	1.21474588404	2.36842105263	1.77777777778	2.00847457627
Mean	0.615013794398	0.730428601148	0.720198106783	0.701785096759
Min	0.210526315789	0.354285714286	0.29296875	0.277777777778

As for most of the classes the mean aspect ratio is in the area of around 0.7 and for ease of processing it would make sense to have all images in one AR then all pictures are converted to the AR = 0.7.

Step 2 - Compression & Down-Sizing

Additionally the images are also down-sized so that they take significantly less space but preserve enough of level of detail to be able to classify them correctly. The selected final resolution is 210x300 pixels (it corresponds ot the Aspect Ratio of 0.7).

Step 3 - Renaming

For ease of grabbing the pictures later a renaming step is applied where the pictures are sorted and renamed numerically such as 1.jpg, 2.jpg etc.

Note: All these processing steps are present in the jupyter notebook: "Capstone Part 01 - Dataset Preparation & Exploration.ipynb"

Main observations about the dataset:

- aircraft class

- contains two aircaft type of data with one airliner Boeing 737 and one general aviaiton plane Cessna 172. This is thought to be enough to allow model for generalization to any aircraft type.

- backgrounds of aircraft pictures are mixed between ground, sky, runways etc. however all pictures in the dataset are presenting aircraft in flight. As the system for object detection is thought to be used only in flight this should correspond to the actual real use case for aircraft detection.

- most aircraft in the pictures occupy most of the frame, in few cases after the reshaping and aspect ratio changes the aircraft are even cropped but not by much. This should create a good dataset for training of actual aircraft features for future detection.

- painings of aircraft are different and should allow for good generalization.

- birds class

- contains all kinds of birds - this means high diversity in texture, shapes, winds positions, sizes etc.

- birds images are pretty much all on the shape of a clean blue sky which can create problems for the future given that there is a separate sky category and their proximity can lead to misclassification

- birds are mostly in a center position within the frame which is a positive thing given the fact that some cropping or Aspect Ratio modifications can be applied later.

- sizes of birds within the frame are quite varied and actually most of the birds fill up only a small part of the frame. This could be lead potentially to some problems for the ML models in training and infrance. More specifically this can lead to:

- classification as sky due to the fact that most of the picture area is occupied by sky class or,

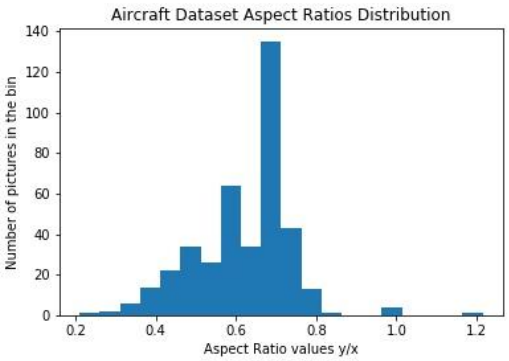
- this can lead to classyfing anything that is relatively small within a frame with sky background as a bird. In this case an aircarft could be potentially mis-classified as a bird just because it is relatively small within a frame.

Note: potential solution to this problem could be a better dataset containing already either bounding boxes around specific objects, perhaps some object segmentation techniques such as morphological segmentation or finally other object localization techniques could be used within a frame.

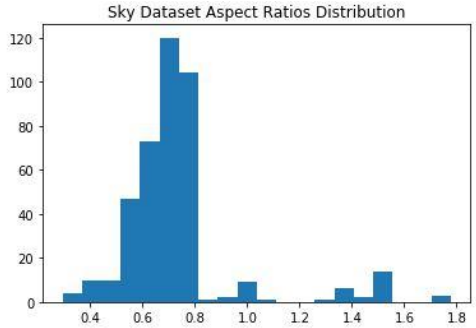
Exploratory Visualization

For the purposes of trully exploring the effects of Aspect Ratio conversion as described in the Step 1 of the Data Exploration chapter the visualisations are made representing the actual distribution of Aspect Ratios per each of the classes in the Dataset.

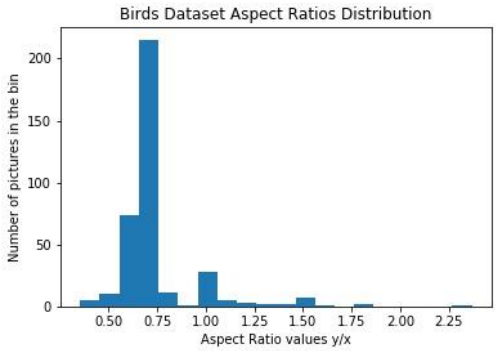
Distribution of Aspect Ratios:



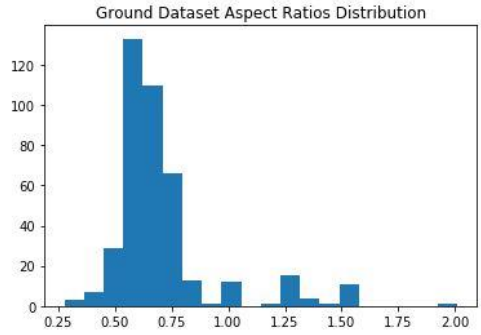
Aircraft Images Aspect Ratios



Sky Images Aspect Ratios



Bird Images Aspect Ratios



Ground Images Aspect Ratios

Observations based on the exploratory visualisations of aspect ratios:

- all classes ARs distributions resemble more or less the gaussian distribution shapes with some alternations regarding how skewed they are, the steepness, the omega etc. For all their peak is around 0.7 which is also indicated by their means.

- some outliers are present in the dataset in terms of Aspect Ratios and ther eare some extremely shaped pictures which are in the portrait mode or extreme landscape shapes.

- for the classes of sky and ground this does not create a problem.

- for the classes of Aircraft the landscape outliers are okay and for the portrait ones after investigation they actually still contain most of the aircraft within the frame so they should stay too.

- for the birds the outliers of Aspect Ratio do not create problems as the birds themselves do not fill out large parts of the frame anyways. The Aspect Ratio conversions therefore will only make the birds fill out more area within the image.

Algorithms and Techniques

In the scope of this capstone project two separate approaches or algorithms are used in order to detect and classify objects within the images.

Approach 1: SVM based on HOG features

Reference jupyter notebook: "Capstone Part 02 - Apply SVM"

Justification of choice: Support Vector Machines are a reliable choice when it comes to Machine Learning and object classificaiton. They work fine with a well defined set of classes which are pretty uniform and pretty distinguishable. This algorithm is selected mainly to serve as a benchmark compared to other more advanced methods.

Input Data handling: Input data needs to be passed through a pipeline extracting specific computer vision features for each image in order to pass it further to SVM. The processing pipeline is based on Histogram of Oriented Gradients HOG features. It is a standard processint technique used in computer vision for object detection. This technique counds occurences of gradient orientation in localized positions of an image. The image itself is divided into multiple bins and HOG features are extracted for each of them. Visualisaiton of this will follow in the next sections.

Default variables & hyper-parameters: HOG feature extraction requires a number of hyper-parameters values to be set and fine-tuned:

- Color spaces for which to calculate Histograms of Oriented Gradients. Possible choices are color spaces of RGB, HSV, LUV, HLS, YUV and finally YCrCb. After some expermentations on randomly selected parameters the YCrCb color space was selected as it seemed to capture the contours of objects and their texture relatively well.

- Orient - number of Orientations Bins for the Histograms of Oriented Gradients. After some finetunning the value of 9 possible HOGs was picked.

- Cells parameters for which to calculate a specific HOG - they include spatial bin size, pixels per cell and cells per block. After some experimentation the final values selected are 15 pixels per cell, 2 cells per block, spatial size of 32x32 and finally 16 histogram bins.

- HOG channels allow to specialize in a specific color channel for the HOG features - can be 0, 1, 2 or ALL. The final choice was made to ALL channels as it seemed to work fine in terms of capturing the relative information about the image as well as was fast enough allowing for quick computation.

Approach 2: CNN based on AlexNet + Transfer Learning

Reference jupyter notebook: "Capstone Part 03 - Apply CNN with Transfer Learning from AlexNet"

Justification of choice: Convolutional Neural Networks are in todays Computer Vision a reference go-to method which is considered to be state-of-the art when it comes to object detection in images. This method was selected to have the most sophisticated final results and hopefully outperform the benchmark method with the Support Vector Machines.

More specifically the AlexNet due to its superior performance on the ImageNet competition for object classificaiton in images. There are today more refined networks also outperforming in terms of accuracy of classification the AlexNet such as VGG, GoogLeNet or ResNet but AlexNet still stands out in terms of being simple to implement and relatively small.

Due to the fact that the training dataset for this particular project is relatively small and contains around 400 images per class and 4 classes it is best practice to use someting called transfer learning meaning to reuse the pre-trained weights already from AlexNet and ImageNet competition dataset. This will allow for only final finetunning and creation of the new layer of the AlexNet and reusing the previously captured weights. In practice this means that the network has already learned to recognize the features on a large ImageNet dataset and now it only needs to learn the final four classes.

This approach seems realistic given 400 images per class and a great performance of AlexNets and transfer learning methodologies in the field of computer vision and deep learning for object recognition.

Input Data handling: For the CNN approach based on the AlexNet and transfer learning there is essentially a re-use of the pre-computed weights from AlexNet ImageNet competition. Additionally in order to make the network work the images need to be resized to the resolution on which AlexNet was trained which is 277x277 and with 3 color channels.

Default variables & hyper-parameters: The default variables and hyper-parameters for CNN with AlexNet mainly concern the network itself. Here for the purposes of this work the AlexNet itself up until the 7th CNN layer was not altered. THis means that parameters for the network layers such as strides, padding, ksize, pooling, activation function are unchanged from the original architecture of the AlexNet. Only the final fully connected layer is added to account for new classes.

Some additional hyperparameters which were used in this work are settings in regards to the batches sizes selected for training and number of epochs. For the purpose of this work there are 10 epochs per batch selected and the batch size is 10.

Benchmark

Benchmark of this capstone project between two methods SVM based on HOG features and CNN AlexNet with Transfer learning is the accuracy of classification of test images. This serves as the main quantitative metric and is used as a main benchmark between the two methods.

In addition there are also qualitative assessments in this work based on the images downloaded from the internet as well as a final vide. They will allow in the end to get a human perspective on the final accuracy of the two methods using a simple visual inspection which can confirm correct classification of objects.

III. Methodology

Data Preprocessing

Data Preprocessing Steps are already largely described in the previous sections of this report. Here is only a quick summary.

Approach 1: SVM with HOG features

Reference jupyter notebook: "Capstone Part 02 - Apply SVM"

Step 1: Resize all the dataset images to the same resolution of 210x300 pixels (giving Aspect Ratio of 0.7)

Step 2: Set all hyper-parameters for the feature extraction for the Histogram of Oriented Gradients (see section for Algorithms and Techniques to see the final values selected)

Step 3: Loop over all dataset images to extract HOG features

Step 4: Assign Label values to each dataset image

Step 5: Split data into training and test dataset with test set being 10% of the overall dataset.

Step 6: Print out the output, accuracy

Optional Steps for manual qualitative check

Step 7: Design a sliding window functionality that will move windows at different scales and overlaps over larger images looking for smaller objects such as a small bird in a much bigger picture

Step 8: After running on test images the sliding windows and classifying each one of them into one of the 4 classes (aircraft, birds, sky, ground) create a heatmap for aircraft class and birds class for each test image

Step 9: Apply thresholds on heatmaps to cut down their values to lower the false alerts and unstabilities of the algorithm.

Step 10: Design bounding boxes around the hotspots on the heatmaps of the aircraft and birds detections

Step 11: Print out the final test pictures with bounding boxes for aircraft and birds classes as well as their fixed heatmaps

Approach 2: CNN based on AlexNet + Transfer Learning

Reference jupyter notebook: "Capstone Part 03 - Apply CNN with Transfer Learning from AlexNet"

Step 1: Load the AlexNet pre-computed weights and assign them to the new networks variables

Step 2: Create an actual AlexNet with the loaded and assign variables and save the network parameters

Step 3: Add the transfer layer to AlexNet by creating fc8 - fully connected layer and save new network

Step 4: Shuffle the dataset into training and validation where the ratio of validation to all is 0.3

Step 5: Train the augmented AlexNet using GradientDescentOptimizer and batched dataset.

Step 6: Estimate the intermediate accuracies for each 50 steps of the batched training process.

Step 7: Run validation process and check the intermediates and the final accuracies of the predictions.

Additionally the same optional steps are now followed as in the Approach 1 (Steps 7 to Step 11) to allow for qualitative inspection on test images. See section above from Approach 1 to check the pipeline.

Step 12: Design a full end-to-end pipeline for image processing and detection of aircraft and birds together with their bounding boxes.

Step 13: Perform a check on a movie to see the detections in a potential real-life situation of aircraft and birds (potential hazards or obstacles) during the flight of an autonomous aircraft.

Implementation

The process, metrics, algorithms, and techniques implemented for the given data are already clearly documented in the previous sections of "Data Preprocessing" and "Algorithms & Techniques".

Complications with regards to the original metrics or techniques that required changing were:

- compared to the initial project proposal the dataset was decreased from over 1000 per class to around 400 per class. The reasons are two-fold, firstly to make the whole project more portable and quicker to implement. Secondly assuming that more advanced techniques such as transfer learning will be used requiring less data it was not necessary to handle the dataset of this size.

- in addition to the reduction of size the dataset was also simplified by reducing flocks of birds images from the bird class dataset. This made the training set simpler and easier to train.

- initially there was a split into 3 classes of aircraft, birds and environment. However after some work with the project it was found easier if the general class of environment is split into two extra classes of sky and ground. This allowed again to have a much cleaner split between these two as they are visually quite different.

- in the initial assumption of the project there was a consideration of a Convolutional Neural Network but after some research the discrepancy is made from the initial proposal by adding transfer learning technique and re-using AlexNet pre-computed weights. This one step has probably the highest impact on the final accuracy of the images classifications with the CNN in this work.

- implementing the transfer learning in tensorflow is a process requiring some more complicated steps and documentations. In the scope of this work tutorials were used from the github repo: https://github.com/samjabrahams/tensorflow-workshop

Refinement

Improvement, refinements and finetunning made upon the algorithms and techniques used in the implementation:

- for the approach 1 with SVM and HOG features there was a high degree of fine tunning of hyper-parameters to pick the HOG features. After some experimentation a pragmatic approach was adopted and even though not all parameter space was explored the final parameters values were set. The actual values are present in the chapter of "Algorithms & Techniques".

- in both approaches with SVM and CNN based on Transfer Learning there was a concept of heatmaps introduced. The refinement came from the fact that the actual heatmaps needed to be thresholded to allow for a cancellation of false detections as well as for creation of more tight bounding boxes around objects. The actual values of thesholds were set by experimenting and using trial and error.

- additionally other parameters were set using trial and error and thay have mostly to do with the qualitative checks of test images and for the purposes of the demonstration with the video. The parameters are: scales of moving windows, chaning the windows overlapping parameters and finally picking out the regions that should be masked or in other works selecting the region of interest in which object classificaiton was made.

- final adjustement for the CNN method of adding bounding boxes for birds was made. It was seen during testing that sometimes when there was a small distinct feature or texture on the background of sky the aircraft class was confused with the birds class. The reasons why this happens are described in the sections below. Here the refinement was made to introduce a propritization in the labelling or adding bounding boxes to the test images in the CNN approach. Rule is that if an aircraft is detected within the given bounding box then a bird cannot be detected in the same region. This eliminates these false alerts but it is a finetunning without improving the Machine Learning aspect of the model but only the processing pipeline.

IV. Results

Model Evaluation and Validation

Final models are both quite reasonable in terms of aligning with solution expectations. Both approaches with SVM and CNN performed reasonable well on test dataset accuracy (quantitative result) as well as in the qualitative assessment on new images from the internet and a demo video (qualitative result).

The models results - accuracies are:

- 0.8805 for the Support Vector Machines based on Histogram of Orientation Gradients features

- **0.9916** for the Convolutional Neural Networks based on AlexNet and Transfer Learning from ImageNet dataset

The results for the SVM approach 1 are reasonable with close to 90% of accurate classification, however the results of the CNN approach vastly outperform the SVM model with over 99% of accurate classification of images.

Having performed additional qualitative analysis on the datasets of the internet (20 images) proves the extra robustness of the model to the new unseen data and to the abilitiy of both models to generalize well enough. Most of the classifications are correct with the following results:

Apprach 1 w/ SVM - Final Results on Extra Test Images

	Aircraft (actual)	Non-Aircraft (actual)
Aircraft (predicted)	**7**	**8**
Non-Aircraft (predicted)	**4**	**X**

	Bird (actual)	Non- Bird (actual)
Bird (predicted)	**3**	**4**
Non- Bird (predicted)	**3**	**X**

	Aircraft (actual)	Bird (actual)	Non-Aircraft & Non-Bird (actual)
Aircraft (predicted)	7	0	8
Bird (predicted)	2	3	2
Non-Aircraft & Non-Bird (predicted)	4	3	X

Apprach 2 w/ CNN - Final Results on Extra Test Images

	Aircraft (actual)	Non-Aircraft (actual)
Aircraft (predicted)	**11**	**0**
Non-Aircraft (predicted)	**0**	**X**

	Bird (actual)	Non- Bird (actual)
Bird (predicted)	**3**	**1**
Non- Bird (predicted)	**3**	**X**

	Aircraft (actual)	Bird (actual)	Non-Aircraft & Non-Bird (actual)
Aircraft (predicted)	11	0	0
Bird (predicted)	0	3	1
Non-Aircraft & Non-Bird (predicted)	0	3	X

Using these test images various hyper-parameters were tested of sliding windows, overlaps, different SVM features hyperparameters and general observations were found to be quite similar for both models:

- in general if an image contains clear sky and a small countour or texture feature element the it is often classified as a bird. Example of that was seen few times in the test images of the internet analysis. In reality this looks that if very often sky image itself with a small dark distinct feature can be misclassified as a bird. This looks that a bird class is too close to the sky class in terms of semantic content of the image itself and especially given the fact that a bird does not fill in enough of the frame of the image. This is true to both models. For the future an extra processing step should be considered for the purposes of creating a more appropriate dataset where either a boundary of a training image would be tighly sorrounding birds in images or perhaps some kind of image segmentation should be applied to discard this effect. In this work extra processing layer was added with thresholding a heatmap but this is not a machine learning solution but a rather computer vision based. Models themselves however seem to be working find and it is probably a lack of appropriate dataset for birds class and too close classes between birds and sky.

- another shortcoming of both models after qualitative analysis is the fact that if an aircraft appeared small in an image and it was relatively difficult to see its distinct features such as engines, landing gear or texture the model would often classify it as a bird. This is probably especially true due to the fact that birds dataset often contained images of birds not filling the full frame of the image hence the CNN learned that if there is a relatively small object on a sky background then it is probably a bird. This problem is only partially fixed in the CNN pipeline by prioritizing labelling of objects based on the heatmaps and hence if an aircraft and a bird is detected on the same part of an image only aircraft should be marked with a bounding box. This is however a workaround and an appropriate solution should be considered with improved dataset of images or perhaps also using the actual probabilities of each object classificaiton.

Approach 2 Transfer learning model based on AlexNet was selected and derived due to its state-of-the performance capabilities for object classification and the possibility of re-using the pre-learned model weights and just finetunning it on a relatively small subset of data for new object classes classification.

In order to test robustness and sensitivity of the model for AlexNet Transfer Learning approach there were 20 images downloaded from internet containing each of four classes in total but with some alterations as to test the robustness. Some alterations were that the objects were relatively small in the frame of an image or that they were not positioned in the center, sometimes there were few objects of the same class such as aircraft of birds in one image and in one case they were of different scales in differnt parts of the picture.

Just classifying an entire image here with the CNN model would probably not be a good idea so a post processing pipeline was added to scan image via sliding windows at different scales and creating a heatmap of different classes detections. Later on that heatmap was a subject to applying a threshold and then bounding boxes.

The final results of precision and recall show the strengths of the model as well as some of its weakenesses.

Strengths:

- very strong classification of aircraft in images when they are of not very small size (Precision: 1 and Recall: 1)

- birds classification at a reasonable level (Precision: 0.75 and Recall: 0.5)

- not a lot of false positives for birds and aircraft (False Positives: 1 bird and 0 for aircraft)

- good approximation of location of objects with the bounding boxes largely fitting closely around objects

- correct classification of sky and ground (altough their classifications and confusion matrices are not fully evaluated in the qualitative analysis on the test images)

Weakenesses:

- some birds detections missed (False Negatives: 3 birds missed)

- when aircraft is relatively small in an image or when a sliding window sees only part of the aircraft within the frame then aircraft feature become almost undistinguishable from a bird then it gets sometimes classified as such. This was corrected in the postprocessing pipeline this is why there is no false misclassificaiton in the final result but before

thresholding of the heatmaps and prioritization of detection of aircraft over birds such cases are visible in the heatmaps.

- as birds in the training dataset are relatively small within the frame, the rest of the frame is often filled-in with sky background and the bird itself has no bounding boxes and is not localized within the image this created some weak spots for the model later on. Whenever the model saw an element of sky with something with some either sharp edges or lines or an element of a dark texture it would often classify it as a bird. This results again in a number of false classification cases but they are later on filtered out by the postprocessing heatmap. Example of this is a rope which in one of the images is stretched in the middle of the frame on the sky background which produces in the end a False Positive classification with a bird label.

The above results show that generally speaking the model is robust, it performs well in its sensitivity analysis and generalizes well to the unseen data. The reults are very reasonable with bounding boxes quite closely sourrounding the objects and with few misclassificaitons. The changes in the input data affect the final results in a small amount and the results can be still trusted. One needs to remember however that the qualitative results are partially seen after the postprocessing pipeline and are a subject to a qualitative human analysis.

Justification

The models results - accuracies are:

- 0.8805 for the Support Vector Machines based on Histogram of Orientation Gradients features

- 0.9916 for the Convolutional Neural Networks based on AlexNet and Transfer Learning from ImageNet dataset

Approach 1 - SVM based on HOG features:

	Aircraft	**Bird**
Precision	0,47	0,43
Recall	0,64	0,5

Approach 2 - Transfer Learning applied on AlexNet CNN:

	Aircraft	**Bird**
Precision	1	0,75
Recall	1	0,5

The final results indicate that the initial model with approach 1 underperforms compared to model 2 in terms of Precision and Recall results. The main differences are:

- for aircraft model 2 has much higher precision and recall than model 1

- for birds model 2 has higher precision and the same recall as model 1

Additionally from the visual inspection it is visible that the model 2 performs much better in terms of localization of objects in the image and creatiion of thight bounding boxes. Model 1 shows somewhat scattered bounding boxes and sometimes even difficult ones to assess whether they should classify as a correct detection or misclassification. This made the final qualitative analysis of model 1 difficult.

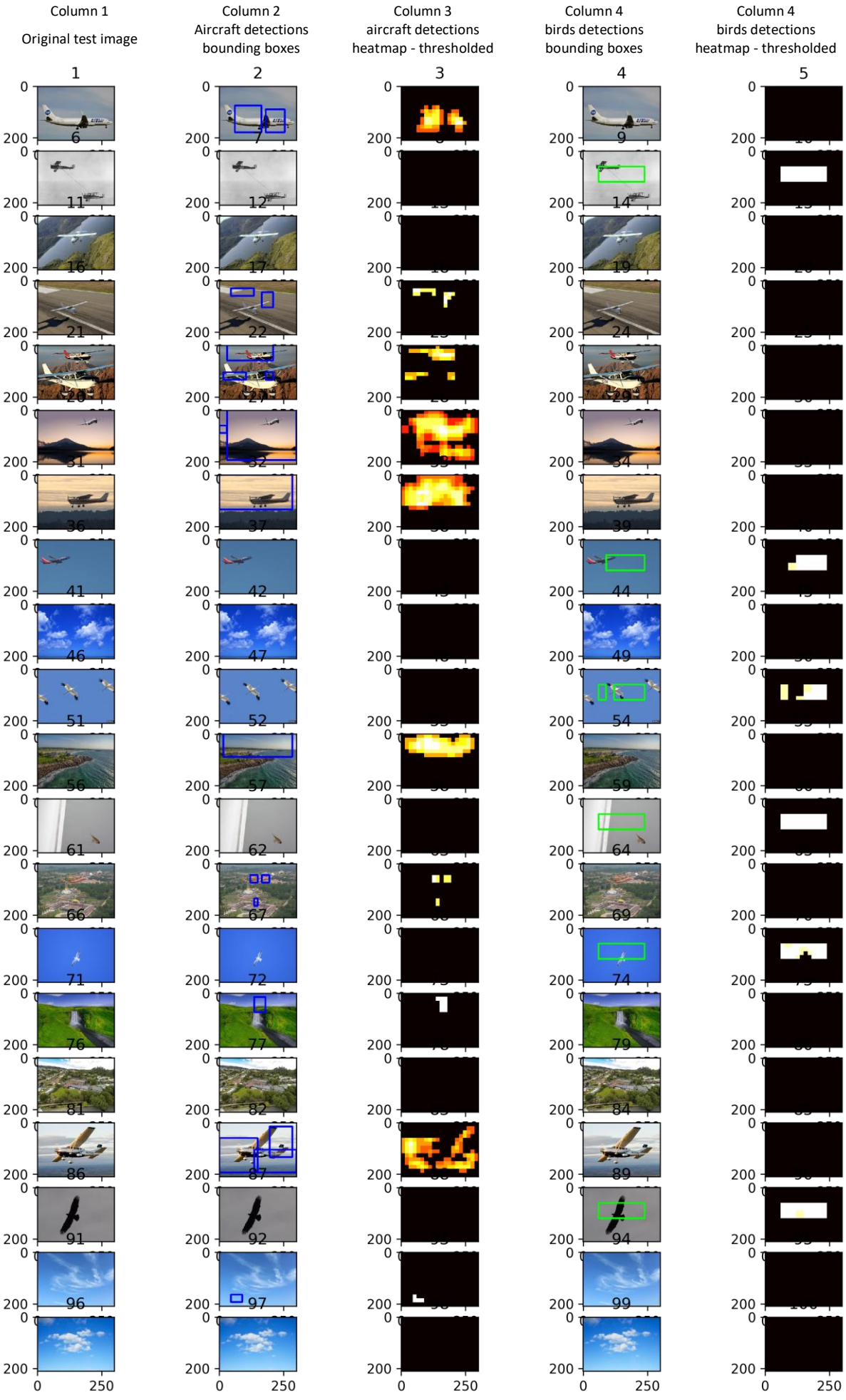
The final solution with approach 2 and transfer learning applied on AlexNet even though clearly outperforming the benchmark solution 1 with SVMs should be still improved. This is true especially due to its weaknesses described in the section on Model Evaluation and Validation and problems with its birds class which sometimes can classify small airplanes and sometimes it also includes a lot of sky. This created the actual precision and recall problems for the bird class. Although the results are reliable and show relatively good results but improvements should be continued to fix the described issue.

V. Conclusion

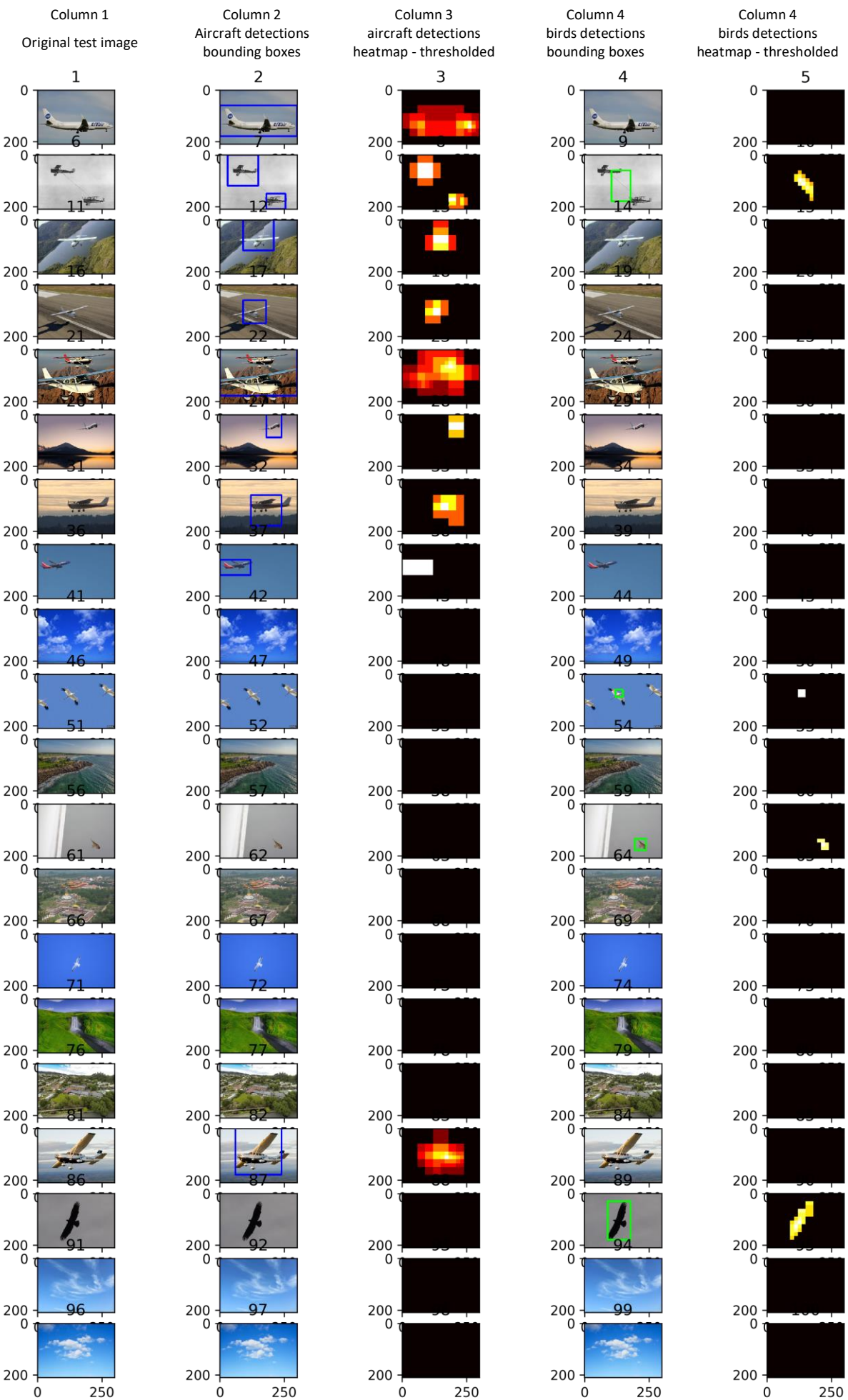
Free-Form Visualization

The visualisations shown below show for model 1 and model 2.

Approach 1 - SVM based on HOG features:



Approach 2 - Transfer Learning applied on AlexNet CNN:


Column 1
Original test image
Column 2
Aircraft detections
bounding boxes
Column 3
aircraft detections
heatmap - thresholded
Column 4
birds detections
bounding boxes
Column 4
birds detections
heatmap - thresholded
1
2
3
4
5
0
200
250

The visualisations above represent the qualitative analysis results, they help to understand the robustness and sensitivity analysis as well as the precision and recall for both models for classes of aircraft and birds. The full description of these phenomenons is included in the above sections as they correspond to these visualisations.

Reflection

Summary and end-to-end description:

The problem that this capstone project addresses is detection and classification of objects from camera images during flight for autonomous flying vehicle that could potentially be hazardous to the safety of its flight. Dataset of around 400 images was gathered of the internet for each of the classes of aircraft, bird, sky and ground. These images were later on used for training of two models of which one served as a benchmark for the final soltion.

The first approach was based on the Support Vector Machines with features extracted from the images using Histograms of Oriented Gradients algorithms. The second approach focused on re-using of pretrained model on ImageNet dataset using so-called AlexNet based on Convolutional Neural Networks and then adding an extra layer to the network and retraining the model just on the gathered dataset for the new classes.

Both of the models then gave their accuracies of around 88% for the SVM and around 99% for the transfer learning approach with CNN on the test images randomly taken from the collected dataset. The final results show superior performance on the test images of the CNN model with almost all correct classifications.

However to allow for an additional analysis there were 20 images collected from the internet containing all classes, with some tweaked examples such as few objects in one image or different scales of objects. In order to deal with additional complexity of the images a pipeline was designed that uses a sliding windows approach at different scale to classify parts of the image into one of the four classes. Then the detections create heatmaps for each of the class and some filtering and thresholding is applied on the heatmaps to clean up the classifications but also to create bounding boxes to localize images within the image frame.

Final analysis of the additional images showed the results for the second model outperformed the benchmark as well but indicated few weakenesses of the solution too. If aircraft are small within the frame and its distinct features are not distinguishable then it can get misclassified as a bird and also sky was sometimes classified as a bird due to high amount of sky backgrounds in the birds' images training set. After applying the pipeline the aircraft results were precision 1 and recall 1 and for birds it was precision 0.75 and recall 0.5 which is still satisfactory but not perfect.

Interesting aspect:

One observation for me during the work that was really eye opening was the ease to take already pretrained model and just finetune it to the new classes on actually relatively few examples. This shows that applying CNN with a really well trained model on a relatively large dataset contains a lot of information about general object features which allows really easily to transfer this knowledge to other fields and objects.

Difficult aspect:

Probably the most difficult part of this work was actually preparing and curating a dataset that would be relatively representative for the problem of detection of other aircraft and birds in the sky during the flight and also that would be appropriate to be solved using machine learning and computer vision. The biggest challenge was that even though the images are already present and they contain the objects needed within their frames it is still difficult to precisely localize where they are exactly within the frame coordinate system and also what is their relative size. This later on created some issues in the accuracy on new images and also some lacks of precision and recall and general results of the confusion matrix.

Applicability of the solution to solve the actual real world problem:

Probably this technique should not be used 100% in this format to detect, classify and localize potential obstacles during the flight of an autonomous vehicle. It shows some promising results but it needs to be significantly improved especially its accuracy and false detections which can actually influence safety of a flying vehicle, speed of classification as today the method is relatively expensive to run in real time using little computational power and finally generalization to unseen data needs to be ideal if it is to be deployed as it is now in the real world scenario.

Improvement

Possible improvements to the current approach could include:

- training dataset compilation in which the objects are not only present in the images but also they are localized with their corresponding bounding boxes. This has the potential to improve the precision and recalls of the final model.

- another improvement to the training dataset is that one could include not only the images of the final objects but also parts of the final objects in the images for example part of an aircraft such as wing. THis would also hopefully allow for better generalization of the algorithm.

- post-processing pipeline that is used today could be also improved to make it faster and to result in more accurate heatmaps or perhaps take a completely different approach to postprocessing to detect and localize objects.

- different network architectures could be used for transfer learning which are perhaps even more capable than AlexNet for object classificaiton. Some examples include VGG, GoogLeNet and ResNet.

- completely different new techniques could be employed to solve this problem which have higher potential. One of those techniques is so called YOLO (You Look Only Once) which is known to have superior object classification and localization capabilities in images and also is extremely computationaly efficient and can be used in real-time systems. It could potentially outperform the solution used in this capstone project.