

Drone-Assisted Inspection for Automated Accident Damage Estimation: A Deep Learning Approach

*M Adel Serhani, *Tony T. Ng, **Asma Al Falasi, *Meera Al Saedi, *Fatima Al Nuaimi, *Hamda Al Shamsi, *Al Damani Al Shamsi

*College of Information Technology, UAE University
Al Ain, United Arab Emirates

**Al Ain City Municipality
Al Ain, United Arab Emirates

Emails: *{serhanim, tonyng,201309306, 201301120, 201331492, 201110150}@uaeu.ac.ae, **asma.alfalasi@am.ae

Abstract—Drones have been used in many application domains nowadays including traffic congestion control, weather information collection, disaster and rescue interventions, and surveillance operations. The drone adoption lies on their capabilities to collect images, videos as well as other sensory data from the air, stream this data to the cloud for processing, and analytics in order to derive important real-time decisions. In this paper, we propose a drone assisted inspection for accident damage estimation based on deep learning approach. Drones are automatically scheduled to visit the accident locations, and data is retrieved for further processing and analytics. We developed a two-phases damage estimation approach, where in the first phase we use deep learning approach to identify and classify objects from accident's images, and in the second phase we measure the size of damaged objects and we estimate the overall cost of the accident's damages. We evaluated our two-phase approach using data of various accidents, and the classification accuracy we have obtained vary between 0.79 and 0.94 and the accident's damage cost estimation most of time is 100% accepted by the expert.

Keywords—*Damage inspection; estimation; classification; drone; deep learning; object image extraction; size measurement.*

I. INTRODUCTION AND RELATED WORK

Drone is commonly known as an unmanned aerial vehicle (UAV), is an aircraft with no human on board to operate and pilot. UAVs include a drone, a controller, and a communications system. The flight of UAVs may operate with various degrees of autonomy: either under remote control by a human operator or autonomously by onboard computers [1]. UAVs were initially used for dedicated sometimes risky missions such as espionage, military attacks, rescue operations, monitoring, and enemies site exploration [2][3]. Their usage was rapidly expanded to cover many commercial areas including scientific, tourism, agriculture, policing, deliveries, and many other applications [3][4]. With advances in UAVs hardware and software technologies including high-end cameras, intelligent

sensors, integrated GPS, autonomous navigation, longer battery, it becomes possible to implement very sophisticated and intelligent operations within the drone that can completely replace human and provide a high level of accuracy and efficiency.

The traffic accidents phenomenon has spread dramatically in these days, either because of a fault of the driver or a defect in the roads and bridges. These are due to lack of focus or use of the phone during driving the car, recklessness in driving and speeding, lack of commitment to traffic signals or violation of driving signs and rules. These incidents cause damage to human, material, or both. For instance, the average of traffic accidents, which occurred in 2016 in UAE reached 4788 accidents, compared to 4796 incidents in 2015 [5]. Therefore, these damages need to be repaired quickly so that they do not cause other accidents.

There are very few research and applications that have been conducted to use drones for damage inspection and estimate the related cost. These applications include powerline inspection, pipeline inspection, flare inspection, volume calculation, and site management. A project developed within the British ecological society [5] used drones to estimate crop damage by wild boar. The drone takes aerial photographs of agricultural fields, pictures are analyzed through a developed algorithm to identify the damaged areas. Another project of Ragone's homeowners insurer company [6] used drones to inspect storm damage. Rather than climbing on the roof, an insurance company representative pilots a drone that takes pictures and video while the operator stays safely on the ground. Other research projects used drones for automatic long linear infrastructure (power lines) damage inspection. Some of them used deep learning approaches [7], others used supervised classification approach for power line inspection [8] while others used convolutional neural networks for image classification [9]. However, all these initiatives remain at the damage detection level and do not estimate the cost of the inspected damages. Our proposed solution goes beyond damage detection and classification to provide also a cost of damaged objects. This combined deep learning approach for accurate object extraction and classification, and image processing techniques for object size measurement.

In this direction, Al Ain Municipality exhibits a need to automate a damage inspection service that is used to be completed by an engineer who has to travel to accident locations, take photos, inspect damages and estimate the costs. Considering that areas under the jurisdiction of Al Ain Municipality span over 15,000 km² and with the limited number of municipality engineers, this process is normally conducted in one week up to one month and needs to be communicated to other third parties (e.g. Insurance companies, Abu Dhabi Police) which may involve some other iterations to validate and maybe re-estimate the accident damage. This process is lacking flexibility, prone to errors, time consuming, and requires iterations and communication among different stakeholders. The idea of automating this process while reducing the human intervention, reducing process time, integrating the service, and making it available to other stakeholders is the main objective of this project. In addition, to improve the efficiency of the damage inspection visits and the efficiency of estimating the value of damages the following are some key features of our proposed solution:

- Minimize the required time to survey the accident site for any damages to municipal assets

- Provide comprehensive footage of damages and automated analysis of damages.
- Assuring site reachability in case of inspecting crowded sites, or damages made to elevated municipal assets.
- Provide automated damage estimation report to be reviewed by an expert.
- Eliminate human factors that might affect the evaluation processes, like: workload pressure, stress, or interpersonal conflicts.

II. SYSTEM OVERVIEW

In this section, we detail the architecture we have put forwards to automate the accident damage estimation. We first overview the main component and their main roles and then we detail the knowledge base construction and validation.

A. Overview and main components

Figure 1 depicts the overall architecture along with its main components. It incorporates different stages starting from data collection using drone, followed by image extraction and processing, damage object isolation and measurement, training and classification, then finally damage cost estimation.

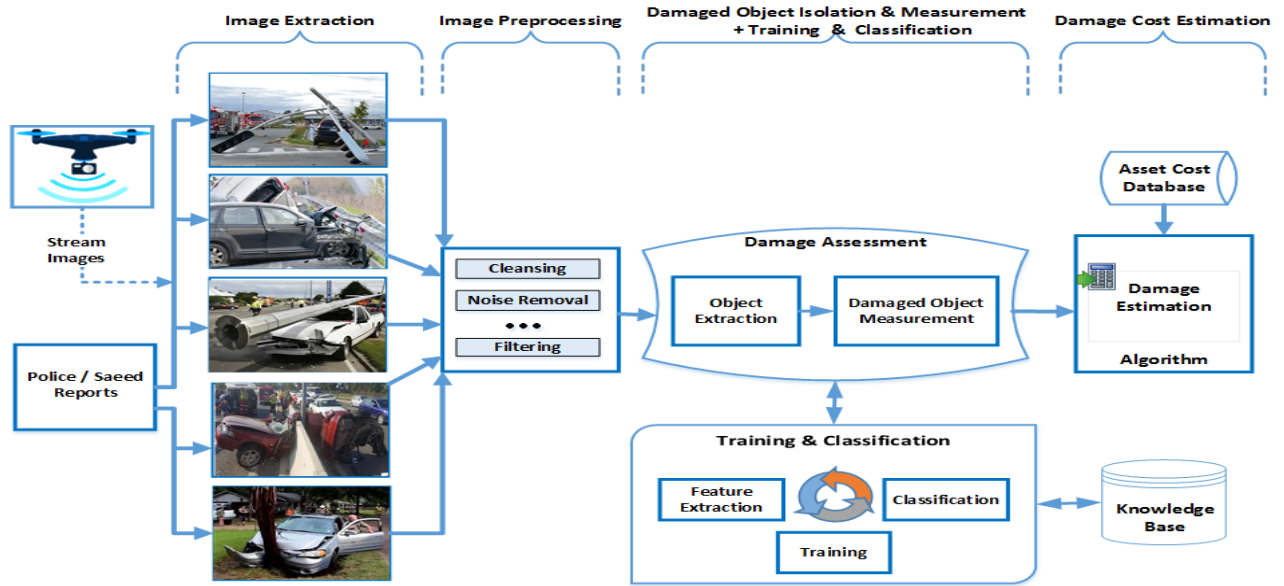


Fig. 1. System Architecture

Drone navigation and data collection: it consists of managing activities of drone navigation, this includes for instance initiating drone voyage from source to the GPS location, automating the drone path and re-routing, and landing the drone due to bad weather conditions. This process also handles data collection, data filtering and preparation. Data consists of pictures and videos, as well as other data recorded about the accident from the Police accident report.

Image processing: it consists of pre-processing, compiling, filtering and analyzing images collected from the drone to conduct image classification, object detection, feature extraction

and analysis to be used in the next process to isolate and measure object's dimensions.

Damage object isolation and measurement: this process involves isolating objects from the images and measuring their dimensions. Different object detection, extraction and isolation algorithms will be used and experimented, this will support the cost estimation of damaged objects extracted from images.

Training and classification: it consists of applying different classification techniques such as artificial neural networks (ANN) to build a classification model, train the model on a set

of images and then predict which class the image belongs to, this will help in the damage cost estimation hereafter.

Damage cost estimation: using inputs from the previous two processes, a cost estimation algorithm is developed to estimate and aggregate cost of different objects involved in the damage. It consists of matching the object extracted from the pictures to measure the degree of damage and the cost associated with different damages. The matching algorithm relies on a pre-established list of common damages stored in an asset cost database. This database is continuously updated whenever the cost of different assets are updated.

The Drone inspection system components are executed in sequence, in a way that the output of a process is delivered to the next process in the chain. For instance, cost of damaged objects are estimated after they are first extracted and their sizes are measured. To allow a smooth integration of all processes in a complete system a set of interfaces were developed to allow smooth integration of different system's components. We also developed applications (Mobile and Web application) to provide access to the system features and provide different users with variety of services offered by the drone inspection system (e.g. automated drone scheduling). Examples of mobile application interfaces are detailed in the implementation system section.

B. Knowledge base construction and validation

Deep learning is a subset of machine learning, which uses ANN to provide fast and efficient results in the classification of big data sets. The advances in deep learning made tasks such as image/object recognition and classification achieving much higher success rates [12].

In this paper, to automate and simplify the object recognition and classification, we used Google TensorFlow, one of the most popular deep learning libraries with Google's pre-trained ImageNet dataset to classify a list of common damage objects in road accidents. Table 1 depicts examples of different public assets along with their prices, measurement units, color, shape, and texture. This information is relevant for object measurement and cost estimation.

TABLE I. LIST OF COMMON DAMAGE OBJECTS IN ROAD ACCIDENTS

Item	Price in AED	UNIT	Department	Color	Shape	Texture
Traffic Signal Pole	270	item	Roads	Black	Cylindrical	Smooth
Palm tree	500	item	Parks	Green/Brown	Cylindrical/Large green leave	Bump/Rough
Metal Fence Pole	280	item	Roads	Green/ grey	Cylindrical	Smooth
Traffic signage Pole	225	item	Roads	Grey	Cylindrical	Smooth
Light Pole	2500	item	Roads	Grey	Cylindrical	Smooth
kerbstone	10	Meter	Roads	Grey/yellow/White	long, narrow block	Rough
Flat Grass area	50	Meter	Parks	Green	Flat surface	Rough
Plants Fence	30	Meter	Parks	Green	Vertical wall	Rough
Flower bed	50	Meter	Parks	Multicolor	Flat surface	Rough

In the ImageNet dataset, there are millions of images belonging to 1000+ different classes, which includes training images, validation images, and testing images [11][13].

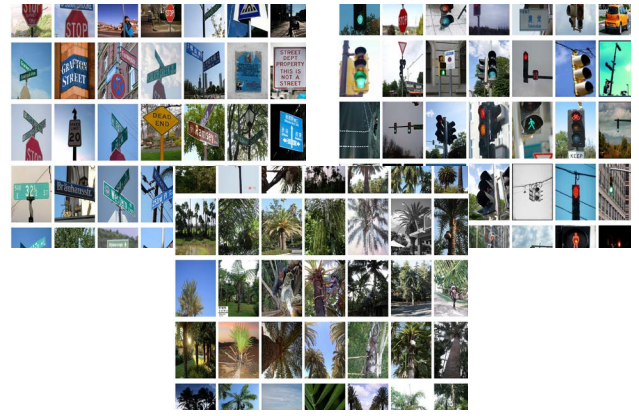


Fig. 2. Sample of ImageNet dataset

To improve the accuracy of image classification, one of the most popular techniques used is convolutional neural networks (CNN). By using CNN in deep learning, more accurate predictions can be achieved in classification approaches [12].

In this research, GoogLeNet Inception-v3 CNN model was used for the classification prediction because of its 3.46% "Top-5 Error Rate", which refers to how often the model fails to predict the correct answer as one of their top 5 guesses. This accuracy could match or exceed human performance in some domains [13]. According to the blog post [14] by Andrej Karpathy who measured his own performance. He only achieved 5.1% "Top-5 Error Rate".

III. DAMAGE ISOLATION AND OBJECT DIMENSION MEASUREMENT USING DEEP LEARNING

In this section, we describe the damaged object classification using the CNN and the object dimension measurement using an open source computer vision and machine learning software library, OpenCV.

A. Damage object isolation and classification

CNN in Figure 3 classifies an input image into four classes: Signage Pole, Signal Pole, Palm Tree, and Fence. On receiving a signage image as input, the network correctly assigns the highest probability for Signage Pole (0.84) among all four classes. The sum of all probabilities in the output predictions should be one.

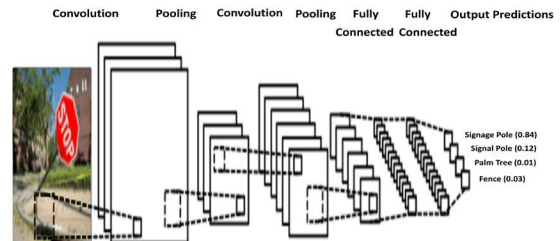


Fig. 3. A simple CNN [13]

Our classification process takes an image captured by drone from an accident site and runs it through TensorFlow Inception-v3 model [15], which compares with the pre-trained dataset, then predicts the probability (0.0-1.0) in which class the image fits.

In the following, we describe the main steps we have implemented to build, train and test our CNN:

Step 1 - Collecting the Dataset: we used Google's pre-trained ImageNet dataset which includes millions of images of 1000+ different classes.

Step 2 - Importing Libraries and Splitting the Dataset: the dataset is already split into training images, validation images, and testing images.

Step 3 - Building the CNN: it consists of three parts: convolution, pooling and flattening. Convolution extracts features from the input images. Every image is a matrix of pixel known as feature map. We use *ReLU* as activation function after every convolution operation. For pooling the feature map is reduced using max pooling with a special neighborhood of 3x3 window and the average pooling of all elements in each matrix is taken. Finally, in flattening the matrix is converted into a linear array that is given as input into nodes of our neural network.

Step 4 - Fully connected: CNN is connected to a neural network, then fully connected network is compiled. Our network is made of 2 layers neural network and we use a sigmoid function as an activation function for the last layer to calculate the likelihood of the four objects: signage pole, signal pole, palm tree, and fence.

Step 5 - Training our Network: after the model is constructed we train our network on the training ImageNet dataset.

Step 6 - Testing: we evaluate our network on the ImageNet test dataset and the prediction results predicted the four objects while the prediction accuracy is relatively high. Different methods can be used to evaluate the performance of the prediction model, for example, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean absolute percentage error (MAPE). However, the more we add convolutional and pooling layers, increase the number of nodes and epochs might improve the prediction accuracy. We used MAE to evaluate the performance of our CNN model. The following formula define the MAE metric:

$$MAE = \sum_{t=1}^n \frac{|y_{pt} - y_t|}{n} \quad (1)$$

where y_{pt} is the predicted value and y_t is the actual value and n is the number of data entries.

B. Object dimension measurement

There are several techniques used for object dimension measurements in the literature. These include for instance triangulation, edge and contour detection, and angles images scale. In this work, we use contour detection to measure object's width or height based on pixels calculation and we implemented

it using OpenCV to determine the size of damaged objects in meters.

To find the size of an object from images using OpenCV, we first must calibrate the drone camera to minimize distortion or inaccuracy of photogrammetry in the images captured by the camera. With the calibration you may determine the relation between the camera's natural units (pixels) and the real-world units (e.g. inches or meters). This is also known as the "pixels per metric" ratio [17].

By using standard sedan car's height (or width) as a reference object to define our pixels-per-metric, which is defined as:

$$\text{pixels-per-metric} = \text{object-height-pixel} / \text{known-height}$$

An average sedan car has a known-height of 1.45 meters. Suppose that car object in the image has an object-height-pixel (measured in pixels) of 200 pixels.

The pixels-per-metric is therefore:

$$\text{pixels-per-metric} = 200 \text{ px} / 1.45 \text{ m} = 138 \text{ px}$$

Thus, implying there are approximately 138 pixels in our images representing every meter of measurement in real-world objects. Then by using this ratio, we can compute the size of damaged objects in the drone images. The two images shown in the figure 4 depict the light pole and sign pole objects extracted from the two images and their size measured in meter.

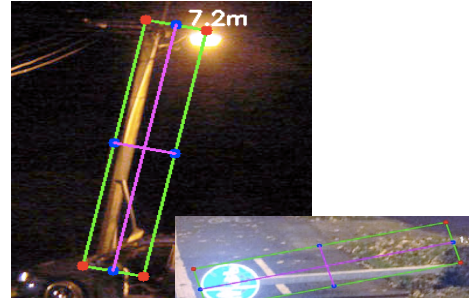


Fig. 4. Object size measurement example

IV. OBJECT EXTRACTIN AND COST ESTIMATION ALGORITHMS

Damage cost estimation consists of three main activities: extracting damaged objects from images, measuring the size of these objects, estimating the cost of each damaged object. We developed two algorithms, one for object extraction and another one for object size measurement and cost estimation.

A. Algorithms description

We developed two interrelated algorithms, the first algorithm is used to extract objects from images, then the second algorithm is used to estimate the cost of objects extracted by the first algorithm. Object extraction algorithm takes as an input a picture, implements object extraction for each picture, extract for each object its name, size, and dimension. Then saves each object information into a vector of *ObjectList*. Repeats these operations for each picture and finally returns a list of objects

along with their names and sizes. Object extraction and its size measurement implementation from an image can be implemented using different approach and tools. We used OpenCV and Python to implement object extraction within our algorithm.

Algorithm 1 Object Extraction algorithm

```

1: Input: Image // Picture containing multiple objects
2: Output: ObjectsList // vector of objects
3: procedure OBJECTEXTRACTION(Image)
4:   while findObject(Image) do
5:     obj ← createObj(name, size, unit)
6:     addToList(obj, ObjectList)
7:   end while
8:   return ObjectList
9: end procedure

```

Algorithm 2 describes the object damage cost estimation which uses as input the list of extracted objects resulted from algorithm 1, then calculates for each object its corresponding price using the size of the object and the equivalent price per unit of an object stored in a database of object prices. For each object in the *ObjectList* the estimated price is calculate and saved in the *EstimationList*. After iterating on all pictures the algorithm returns a list that contains for each picture, the estimated prices of all its imbedded objects.

Algorithm 2 Object Estimation algorithm

```

1: Input:
   ObjectsList // vector of objects
   EvalDatabase // Database of evaluation method for each object type
2: Output: EstimationList // vector of Estimations
3: procedure OBJECTESTIMATION(ObjectsList, EvalDatabase)
4:   for all obj ∈ ObjectsList do
5:     pricePerUnit ← getUnitPrice(obj.getUnit(), EvalDatabase)
6:     objEst ← pricePerUnit × obj.getSize()
7:     addToList(objEst, EstimationList)
8:   end for
9:   return EstimationList
10: end procedure

```

V. IMPLEMENTATION AND EVALUATION

In this section, we first describe the implementation of a system we have developed to manage the drone navigation, schedule drone visits to accidents' locations, data collection (pictures and videos), real-time monitoring, autonomous flight, data download, and final data analysis for damage estimation. We then, describe image processing, classification and knowledge-based implementation. The drone navigation system usually consists of 3 sub-systems; front-end system, back-end system and data analysis system.

To develop different components of our drone inspector system, we have used the following software and hardware systems:

Hardware	Software
<ul style="list-style-type: none"> Laptop Smart device Drone 	<ul style="list-style-type: none"> TensorFlow Python OpenCV SQL DB Web services Map/Location based system

A. Drone Inspector Mobile Application

This is a mobile/web application that is directly accessed and interacted with by the user to schedule, monitor, or control the drone while in the air and take the necessary actions when needed. Such actions include, for instance, automatic "Return Home" for the drone because of weather condition or battery drainage. Integration of real-time drone navigation and map/location information is also visualized. This application is developed using common programming languages such as Java, JavaScript, Visual dotNet, PHP, HTML5, or XML. The following, are examples of mobile application's interfaces:



Fig. 5. Sample drone inspector mobile application's interfaces

B. Drone application engine

This is also known as ground control system (GCS) in drone technology. This GCS is the main system that provides and supports the drone inspector functionalities. It is also responsible for receiving and responding to the front-end system requests and operations. It also contains database and data processing components for data collected from drone and data analysis. Such GCS is developed using common drone programming languages; Python and C++.

C. Data Analysis System

This is an extension of front-end system, which provide the user with damage estimation after data from drone has been downloaded and analyzed. Afterwards, the expert can accept, revise, or reject the estimation. Figure 5 shows an example of objects classification and cost estimation for two objects: a traffic signage pole and signal pole. The traffic signal pole classification score is 0.84 and its estimated damage cost is of 225 AED. However, traffic signal pole classification score is 0.79 and the estimated damage cost is 270 AED.



Traffic Signage Pole	Traffic Signal Pole
	
\$ python test.py --image_file=Dronelmg1.jpg	\$ python test.py --image_file=Dronelmg2.jpg
Traffic Signage Pole (225 AED) (score = 0.84297)	Traffic Signal Pole (270 AED) (score = 0.79662)

Fig. 6. Sample of classification prediction results

VI. CONCLUSION

UAV's were heavily used nowadays to support various domains in handling complex, time consuming and critical situations where human intervention is difficult.

In this paper, we proposed a drone assisted inspection for accident damage estimation using deep learning approach. We automatically schedule drones to fly to the accident locations, retrieve data (e.g. pictures, videos), and relay it to the cloud for further processing and analytics. We developed a two-phases damage estimation approach, wherein the first phase we use deep learning approach to identify and classify objects from accident's images, and in the second phase estimates and measures the cost of the damage caused by the accident. Unique features of this research involve automatic drone inspection visits and monitoring as well as deep analysis of data for accident's damage cost estimation. This developed solution allows saving significantly the time while automating the end-to-end accident damage identification, classification and damage cost estimate.

As future work, are planning to improve the classification accuracy and object measurement while experimenting other classification and damaged object's size measurement techniques. We are planning to build enhance our CNN model by adding more hidden layers, increase the number of nodes and epochs which will definitely improve the prediction accuracy. We are also planning to conduct a comparative study with other benchmarking systems for accident's damage estimation. Finally, we are going to complete the implementation of the full drone inspection system and deploy it on a real environment.

ACKNOWLEDGMENT

This research work is supported by SURE+ research fund # 31T108

REFERENCES

- [1] "ICAO's circular 328 AN/190 : Unmanned Aircraft Systems" ICAO. Retrieved 3 February 2016.
- [2] C. W. Lum, K. Gauksheim, T. Kosel, and T. McGeer, "Assessing and estimating risk of operating unmanned aerial systems in populated areas," in 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 2011.
- [3] B. Blickensderfer, T. J. Buker, S. P. Luxion, B. Lyall, K. Neville, and K. W. Williams, "The design of the uas ground control station: Challenges and solutions for ensuring safe flight in civilian skies," in Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 56, no. 1. Sage Publications, CA, 2012, pp. 51–55.
- [4] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," ACM Transactions on Cyber-Physical Systems, vol. 1, no. 2, p. 7, 2016.
- [5] Sabrina Weiss, "Using drones to estimate crop damage by wild boar", British Ecological Society, 13-Dec-2017. [Online]. Available: <https://www.britishecologicalsociety.org/using-drones-estimate-crop-damage-wild-boar/> [Accessed: 15-Dec-2018].
- [6] B. Marquand, "Meet Your New Claims Inspector: A Drone", NerdWallet, 09-Jun-2017. [Online]. Available: <https://www.nerdwallet.com/blog/insurance/drones-home-insurance-claims-inspectors/> [Accessed: 15-Dec-2018].
- [7] J. Gubbi A. Varghese P. Balamuralidhar "A new deep learning architecture for detection of long linear infrastructure" The 2017 IEEE Proceedings of Machine Vision Applications May 2017.
- [8] C. Sampedro C. Martinez A. Chauhan P. Campoy "A supervised approach to electric tower detection and classification for power line inspection" 2014 International Joint Conference on Neural Networks (IJCNN) pp. 1970-1977 2014.
- [9] A. Krizhevsky I. Sutskever G.E. Hinton F. Pereira C.J.C. Burges L. Bottou K.Q. Weinberger "Imagenet classification with deep convolutional neural networks" in Advances in Neural Information Processing Systems Curran Associates Inc. vol. 25 pp. 1097-1105 2012.
- [10] Emarat Al Youm, 13 Mar. 2017. [Online]. Available: <https://www.emaratalyoum.com> [Accessed: 15-Dec-2018].
- [11] Krizhevsky, Alex & Sutskever, Ilya & E. Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.
- [12] Ertam, Fatih & Aydin, Galip. (2017). Data classification with deep learning using Tensorflow. 755-758. 10.1109/UBMK.2017.8093521.
- [13] An Intuitive Explanation of Convolutional Neural Networks, The Data Science Blog: www.ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- [14] What I learned from competing against a ConvNet on ImageNet, Andrej Karpathy Blog: karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/
- [15] TensorFlow: www.tensorflow.org
- [16] ImageNet: www.image-net.org
- [17] Measuring size of objects in an image with OpenCV, PyImageSearch Blog: www.pyimagesearch.com/2016/03/28/measuring-size-of-objects-in-an-image-with-opencv/