Mobile Application to Identify Fish Species Using YOLO and Convolutional Neural Networks

K. Priyankan#1, T.G.I. Fernando#2

#Department of Computer Science, University of Sri Jayewardenepura
Gangodawila, Nugegoda, 10250, Sri Lanka
[1]priyankankiru@gmail.com [2]gishantha@dscs.sjp.ac.lk

Abstract. Object detection is one of the sub-components of computer vision. With recent development in deep neural networks many day-to-day problems can be solved. One of the practical problems faced by shoppers is the difficulties in identifying the fish species correctly. Even though there are few studies to solve this problem, those implemented solutions are not easily accessible. Main objective of this study is to implement a mobile application based on deep learning that can detect the fish species and provide information on vitamins, minerals, prices and recipes. For this study, top selling 16 Sri Lankan fish species are used. In this study, we were able to build a model using a YOLO based convolutional neural network. Mobile application takes 3-20 seconds to detect the fish species based on the Internet speed.

Keywords: fish detection, convolutional neural network, YOLO, detection and classification.

1 Introduction

People visit the fishing market very often since fish products have vital vitamins and minerals which are required for healthy life. Since the fish products fulfil 53% of animal protein, the per capita consumption the country has achieved 44.6 grams per day [1]. Super market industry offers wide variety of food and household items where customers have many advantages compared to traditional grocery shops. Most of the supermarkets have tags specifying the product details and price details for the convenience of the customer. One of the sections in a supermarket is the fresh fish/meat section.

The current approach followed in buying fish in most of the supermarkets and fish markets is, sliced pieces/whole fish are displayed where customers must ask the shopkeeper and get to know the fish type and price details and buy according to the requirement. This consumes plenty of time and sometimes can lead to false information. Customers can spend less time in fish market if the fish species can easily be identified like the tags available on each product at the supermarkets.

Japan has high life expectancy linked to diet. This is achieved due to proper fish consumption. Some Fish products are used to cure many diseases and leads to healthy

life style. People must be able to get information of vitamins and minerals in each fish species and what type of diseases it can cure. These information must be simple in such a way that the general public must understand and also easily available at any time [2].

Machine learning and deep learning play a major role in computer vision. Computer vision applications are widely used with the advancement in technology and resources. It is used in a simple scenario like character recognition to a complex scenario like self-driving vehicles.

Object detection is one of the sub-components in computer vision that solves many day-to-day problems. Object detection is the process of finding instances of real world objects. Even though human and animals can easily detect objects, it is difficult for a machine to detect objects in a scene. But with the use of computer vision we can now detect objects easily. Although many research in the object detection field have been done, very less have tried in solving the fish detection problem [3]. Fish detection can be considered as one main application of computer vision. Object detection differs from object classification, i.e. object classification shows which object is depicted in the image and object detection shows where the object is in the image. In this study, the main focus is to implement a simple deep learning solution for detecting the fish species accurately.

The aim of this study is to implement a mobile application that can identify and detect fish species and then to view the price details, nutrients and recipes efficiently. The objectives of this research are studying the existing tools and technologies that are applied to classify fish types, studying the deep learning algorithms that can be applied to detect and identify objects and develop a novel method to detect and identify the fish type even if the image contains a sliced fish using deep learning algorithms and develop a mobile application which can detect and identify fish type and then provide the vital information about the fish such as price, nutrients and recipes to cook the fish.

2 RELATED WORK

In the study conducted by Eiji et al. [4], a neural network was developed to identify the fish species in Japan by using reference points. Reference points are characteristic points that are extracted from images of the body surface of the fish using a method that employs the truss protocol. The ratio of specific truss lengths between the 'reference points' relative to the total body are used to compile the dataset and it is used as the network input. The study says that the neural network developed provides higher accuracy in identifying the fish species. But this study was conducted only for 3 fish species. Fish species are identified using the head section and colour of the body section. After using the RGB values the accuracy was enhanced proving that the colour plays major role in identifying the species of the fish.

Another study conducted by Michael Chatzidakis [5], has used a convolutional neural network (CNN) to identify fish species in camera footage. For this study, the researcher has considered only 8 species of fish. This study says that 97.4% accuracy was obtained before overfitting. The pre-trained model has been used for this study

with many augmentation techniques applied to the training dataset. Dropout, weight decay and batch normalization have been effectively used to improve the convergence and decrease the overfitting. But this solution cannot be embedded to a mobile device since this need high-performance computing devices to classify fish species.

Li et al. [6] have done a research on Fast Accurate Fish Detection and Recognition of Underwater Images with Fast R-CNN. In this study, they stated that fast R-CNN is more suitable for underwater fish detection and it's comparatively faster than a CNN. For this study 12 types of fish species are used which are found in Deep Ocean. This architecture takes input as an RGB image and its 2000 regions of interest (RoIs) collected by selective search and produces a distribution over fish classes as well as related bounding-box values. The networks contain five convolutional layers, a RoI pooling layer, two fully connected layers and two sibling layers (a fully connected layer and softmax layer over 12 fish classes plus background class and bounding-box regressors). Necessary response normalization and max pooling layers follow the first and second convolutional layers. The rectified linear unit (ReLU) non-linearity is applied to the output of every convolutional layer and every fully connected layer.

Another study "Automatic Nile Tilapia Fish Classification Approach Using Machine Learning Techniques" identifies one single fish using support vector machines [7]. This study uses Scale Invariant Feature Transform (SIFT) and Speeded up Robust Features (SURF) algorithms to extract features. Furthermore, the study states that the experimental results obtained from the support vector machine algorithm outperformed other machine learning techniques, such as artificial neural networks (ANNs) and k-nearest neighbour (kNN) algorithms, in terms of the overall classification accuracy.

Hnin and Lynn [8] propose an automated species identification system to identify taxonomic characters of species based on specimens. It also provides statistical clues for assisting taxonomists to identify accurate species or review misdiagnosed species. For this system, feature selection is an essential step to effectively reduce data dimensionality. By using combination theory, this system creates the set of attribute pairs to construct the training dataset. And then each attribute pair in training dataset is tested by using two classifiers. Based on the accuracy result of each classifier on attribute pairs and the majority voting of each feature in these attribute pairs, this system selects the most relevant feature set. Finally, this system applied three supervised classifiers to verify the effectiveness of selected features subset.

Salimi et al. [9] introduce a system based on the 'Otolith' contours to identify the fish species with the high classification accuracy. They have identified 14 fish species. Short Time Fourier Transformations (STFT) to extract features of the 'Otolith' contours and then Discriminant Analysis (DA) have been used to classify the fish species from the extracted features. This study states that they were able to get 90% accuracy for nearly all the 14 classes they have used.

How fast a trained model takes to detect an image is really important in practical point-of-view. Most of the above studies are not capable enough to detect fish species quickly, so that they can be implemented in handheld devices. This research is unique and differs from others, since there is no other model that is capable to detect Sri Lankan fish species instantly.

3 METHODOLOGY

According to the report [1] published by the Ministry of fisheries and aquatic resources development Sri Lanka, fish production in Sri Lanka are mainly from marine sector, coastal water, offshore sea water, agriculture and shrimp farming. There are around 90 major species of fish that is consumed by the public. It is very difficult to collect a dataset for all the fish species with a limited time of this undergraduate research project. So, we should have a plan to reduce the number of fish species but also solve the problem faced by general public in identifying fish species. Mainly images were collected from below sectors:

- People who consume fish (General Public)
- People who take part in fish production (Fisherman)
- People who sell fish products (Supermarkets/Fish market owners)

By considering all the information collected from all three sectors, we were able to choose the top 16 fish species that is consumed by general public of Sri Lanka [1]. Fig. 1 depicts the top 16 species selected for this study according to the data provided by three groups of people. The most important task in this research is to have a dataset that can be used to train the model. But there is no such a dataset of fish species of Sri Lanka. So collecting and preparing the dataset play a major role in this study.

Fig. 1 Top 16 Fish Species

3.1 Data Collection and Pre-Processing

Before starting to collect the dataset, we had a plan on how to collect the fish images. Since this study is based on object classification we need plenty of images for each fish species. We should have nearly 800-900 images of each species to get a better accuracy. First, we found out the places that we can get the top 16 fish species we selected. Accordingly the following are the places that images were collected:

- John Keells Distribution Centre (Wattala)
- Fisheries Cooperation (Nawina)
- Fisheries Cooperation (Moratuwa)
- Fish Market (Colombo 06)
- Fish Market (Colombo 04)
- Keells Super Centre (Dehiwala)
- Internet Images

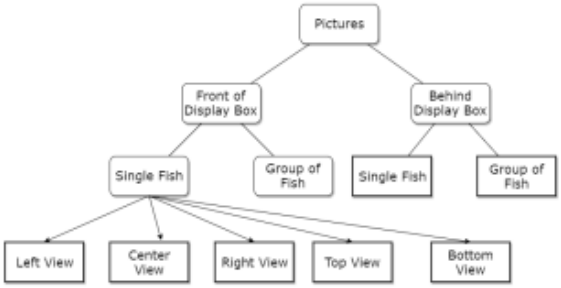

Fig. 2 Variation in taking pictures

Using the plan given in Fig. 2, we collected 120 pictures for each species from each shop and hence we have collected 1920 (=120*16) images. But this is not enough to train a deep neural network. Therefore, we decided to use image augmentation techniques to increase the number of images without distorting the images. The techniques that were used to increase the number of images in the dataset are translation, rotation and flipping.

(a) Group View,Behind Fish Display Box

(b) Single View,Behind Fish Display Box

Fig. 3 Variations in images taken according to the plan

After applying the augmentation methods, the number of images for each class is increased to 150 for each shop. Finally, we have 14,400 images for all top 16 fish species, i.e. 150 (original + images after applying augmentation techniques) * 6 (shops) * 16 (fish species). Fig. 3 shows the variation in images taken for training the model.

This study is not only fish classification but also handles fish detection, where the model should identify where the fish is in the image. To train this kind of a model we need a training dataset with bounding box specifying where the fish is in the image. This process takes a very long period since this must be done manually by the developer one by one specifying the fish species class and bounding box respectively. An open source python program called “BBox” tool [10] has been used to do this process. This tool has a graphical user interface where we can draw the bounding box and then it generates the number of boxes we drew, Xmin, Ymin, width and height of the bounding boxes. Fig. 4 shows the user interface of the tool.

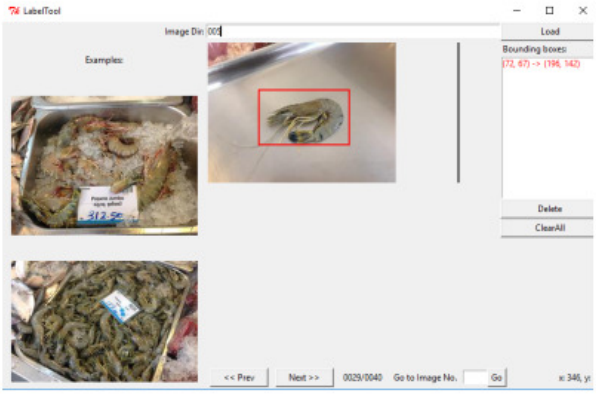

Fig. 4 Pre-processing using the BBox tool

Dataset is the key to all machine learning problems. Number of examples in the dataset affects the accuracy of the model [11]. As discussed earlier only three augmentation techniques were used to increase the number of images in the dataset without distorting images. Only 20 images from each species were taken and translated 5 to 20 pixels up, down, left and right. Similarly, those images were rotated 1 to 2 degrees to left and right. And image flipping technique was also used since flipping fish image does not affect the accuracy of model. Fig. 5 and Fig. 6 depict the images after applying the two augmentation techniques – flipping and rotation.

Fig. 5 Data Augmentation - Flipping

Fig. 6 Data Augmentation - Rotation

After applying these techniques the final dataset was spitted into training and testing datasets as to the ratio 7:3 respectively.

3.2 YOLO

YOLO [12] is a concept called "You Only Look Once." Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections. YOLO applies a single neural network to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities.

This YOLO divides the image into 5*5 grid cells. Each cell is responsible for predicting 2 bounding boxes, which are called the anchors. So in the final layer we get tensor of size 5*5*(5*2+16). Since we have 2 bounding box for each cell, the first 10 values of the 1*26 are:

- X coordinate of the bounding box centre inside the cell.
- Y coordinate of the bounding box centre inside the cell.
- Width of the bounding box
- Height of the bounding box
- Confidence of the class - Pr(Class—object)

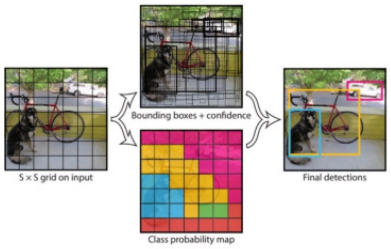

Fig. 7 YOLO Architecture

The remaining 16 cells denote the conditional probability of object belongs to the class i, if an object is present in the box. Next, we multiply the all of these class scores with bounding boxes for each grid cells. So now we have 5*5*2=50 bounding box of (1*16) tensor. Now we use non-maximum suppression algorithm [13] to set score to zero for redundant boxes. After that, it is left with only 2 or 3 bounding boxes where others are set to zero. From these bounding boxes we select boxes according to the class score.

3.3 Convolutional Neural Network (CNN)

A CNN [14]–[17] is a popular deep learning technique for visual recognition tasks because of its proven quality of performance in image classification with less image pre-processing. In machine learning, a CNN is a type of feedforward artificial neural network (Fig. 8). They are widely used in the field of pattern recognition within images and videos. A CNN consists of several layers. These layers are convolutional layers, ReLU layers, pooling layers and fully-connected layers. When these layers are stacked together, CNN architecture has been created. The neurons of the CNN layers are arranged in 3 dimensions (width, height, and no of channels). The no of channel of the input layer is 3 for colour images and it is 1 for gray-scaled images.

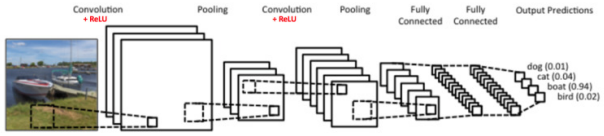

Fig. 8 Convolutional Neural Network (CNN)

Convolutional layer [17] is the first layer in CNN. This layer has a filter/kernel which has weights and biases. The depth of the filter must be same as the depth of the input image. This filter slides over the input image. As we choose the filter size we need to choose the stride and padding. Stride controls how the filter convolves around the input volume. Padding pads, the input volume with the value around the border. As the filter slides around the image, it multiplies the values in the filter with original pixel value of the image and summed up as a single number. This process is repeated for every region of the input image. The final region after sliding is completed the output of this layer is called the feature map.

Purpose of adding a ReLU layers [17] layer after a convolutional layer is to introduce non-linearity since convolutional layers operated with linearity. There are many functions to add non-linearity like 'tanh' and 'sigmoid.' But researchers found that ReLU function performs better than those since model can be trained faster and accurately. ReLU layers changes the negative activation to zero providing non-linearity to the model.

After a ReLU layer, a CNN has a pooling layer. There are many pooling layers like maximum pooling layer, average poling layer and L2 norm pooling layer. Most of the CNN use the maximum pooling layer which takes the maximum value in each sub-region.

Finally, a CNN has fully connected layers, where the input from the previous layer is flattened and sent so as the transform the output into the number of classes as desired by the network.

In addition to that, dropout layers [16] are added between layers to prevent the overfitting problem. Dropout is a technique where randomly selected neurons are ignored during the training.

3.4 Implemented Architecture

YOLO has the advantage over the other CNN. YOLO applies a single CNN for both classification and localization of object. YOLO can process images at about 40-90 FPS. Our network uses 24 convolutional layers followed by 2 fully-connected layers in the first layer. To build this model 90 layers of filters have been used in the second layer. This model takes the input image and resizes to 448*448 pixels. Then the image passes through the first and second layers of network above and outputs 7*7*30 tensor. This output provides the coordinates of the bounding box and probability of the detected class.

3.5 Hardware and software used in the experiment

For this study, we have used the following hardware and software:

- GPU Computer was used with Intel Core i7 CPU, DDR3 16GB of Memory and NVIDIA GeForce GTX 960 processor (2GB) and Tesla K40 GPU.
- Ubuntu 14.04 (64 bit)
- NVIDIA DIGITS 5
- MATLAB R2012
- BBox-Label

4 Results

The accuracy of this model is 77%. Even though there were many studies with better accuracies than this, they require high computing resources or considerable amount of time to detect the fish. Those implemented solutions are not available for easy access to the general public, so that they can solve their problem in identifying the fish species correctly.

The average time taken for detecting the fish species depend on the Internet speed and the quality of the image. On an average this takes 3-20 seconds per image to detect.

Table 1Confusion Matrix

n = 1607	Predicted Yes	Predicted No
Actual Yes	768	202
Actual No	166	471

Table 2 Confusion Matrix Measures

Measure	Value	Derivation
Sensitivity	0.8143	TPR = TP / (TP + FN)
Specificity	0.6999	SPC = TN / (FP + TN)
Precision	0.7828	PPV = TP / (TP + FP)
Negative Predictive Value	0.7394	NPV = TN / (TN + FN)
False Positive Rate	0.3001	FPR = FP / (FP + TN)
False Discovery Rate	0.2172	FDR = FP / (FP + TP)
False Negative Rate	0.1857	FNR = FN / (FN + TP)
Accuracy	0.7652	ACC = (TP + TN) / (P + N)
F1 Score	0.7982	F1 = 2TP / (2TP + FP + FN)

Dataset greatly affects the accuracy and efficiency of a trained model. Table 1 and Table 2 show that model need to be more improved. The accuracy of the model can be further improved by introducing new images for some classes and retrain the model with these images.

A mobile application (see Fig. 9) has been developed to implement the model. Since a mobile device has a limited capacity and speed, a script running on an AWS machine is called from the mobile device when a user input an image for the detection of a fish. Then the output of the application (see Fig. 10) is the detected fish with a bounding box and the accuracy of the model. Further, the user can select the detected fish from a dropdown menu to view the details of vitamins, minerals and price of the fish, and its recipes. The application is also capable to detect sliced fish and multiple fish types in a single image. The model has been trained in a way that the user can input an image taken in any orientation.

Also, a user can send a feedback to the system after it detects a fish correctly or incorrectly. These images and labels are saved in the server so that these images can be used for future training of the model and fine-tune it further.

For more details of the mobile application, a demonstration is available at the YouTube link "https://www.youtube.com/watch?v=PuuBARG4S-0."

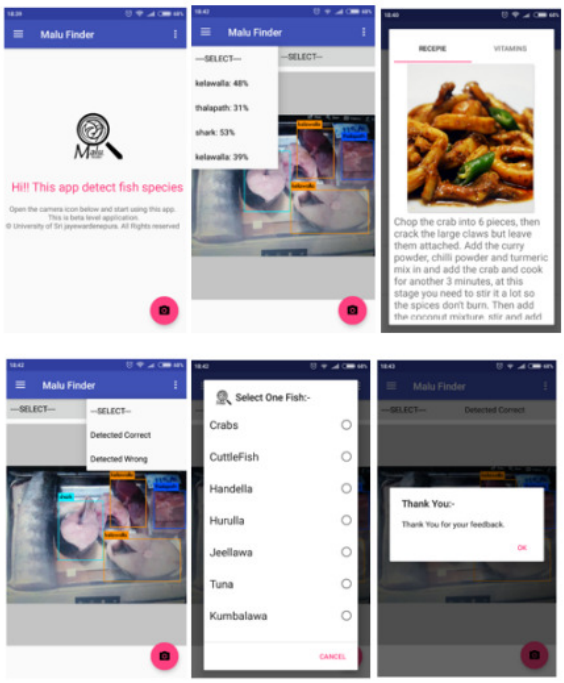

Fig. 9 Mobile interfaces

Fig. 10 Output of the application

5 CONCLUSIONS

As mentioned earlier the main objective of this study is to identify the fish species correctly. Complex part of this study is to identify the sliced fish images correctly. In this study we were able to implement a convolutional neural network using YOLO to identify fish species and locate where the fish in the image. Sliced fish species and images with multiple fish species in a single image also can be classified and detected properly using this neural network.

In this study we were able to review the related work on our problem domain which is fish species detection. Related studies proved that time consumption for identifying with low computational resource is high, but we were able to implement a mobile application that uses this trained neural network successfully. We were able to test the neural network with a reasonable accuracy and low time latency.

CNN requires less image pre-processing compared with other approaches. In this study we can conclude that CNN can be used to detect the fish species with an acceptable accuracy.

We were able to implement a successful mobile application that can detect the fish species and customers can access this service from any place and at any time. But there are several enhancements that can be incorporated to this study.

One enhancement, we added to the application is that a user can send their feedback to system and these feedbacks can be used to fine-tune the model later.

The developed model is tested only for 16 fish species found in Sri Lanka even though there are more than 90 species found. This model can be extended to classify and detect other species as well. And training dataset can be increased to get higher accuracy. Currently the model is hosted in AWS server in which the mobile application access. On an average, this detection takes 3-20 seconds based on the Internet speed. This can be enhanced in a way that model runs inside the mobile application itself, so that we can reduce the time for detection. As another enhancement, the network architecture can be changed and tested in different environment to reduce the training time and also to increase the accuracy.

References

1. “Welcome to the Department of Fisheries and Aquatic Resources.” [Online]. Available: http://www.fisheriesdept.gov.lk/. [Accessed: 23-Mar-2019].
2. “Health & Families,” The Independent. [Online]. Available: http://www.independent.co.uk/life-style/health-and-families. [Accessed: 23-Mar-2019].
3. J. Wu, B. Peng, Z. Huang, and J. Xie, “Research on Computer Vision-Based Object Detection and Classification,” in Computer and Computing Technologies in Agriculture VI, 2013, pp. 183–188.
4. M. Eiji, T. Yuichiro, and N. Makoto, “Identification of Fish Species Using Neural Network,” Journal of National Fisheries University, vol. 58, no. 1, pp. 65–71, 2009.

5. “Using Convolutional Neural Networks to Identify Fish Species in Camera Footage,” Michael Chatzidakis. [Online]. Available: https://www.mikechatzidakis.com/home/2017/7/30/using-convolutional-neural-networks-to-identify-fish-species-in-camera-footage. [Accessed: 23-Mar-2019].
6. X. Li, S. Min, Q. Hongwei, and C. Liansheng, “Fast accurate fish detection and recognition of underwater images with Fast R-CNN,” in OCEANS 2015 - MTS/IEEE Washington, 2015, pp. 1–5.
7. M. M. M. Fouad, H. M. Zawbaa, N. El-Bendary, and A. E. Hassanien, “Automatic Nile Tilapia fish classification approach using machine learning techniques,” in 13th International Conference on Hybrid Intelligent Systems (HIS 2013), 2013, pp. 173–178.
8. T. T. Hnin and K. T. Lynn, “Fish Classification Based on Robust Features Selection Using Machine Learning Techniques,” in Genetic and Evolutionary Computing, 2016, pp. 237–245.
9. N. Salimi, K. H. Loh, S. Kaur Dhillon, and V. C. Chong, “Fully-automated identification of fish species based on otolith contour: using short-time Fourier transform and discriminant analysis (STFT-DA),” PeerJ, vol. 4, Feb. 2016.
10. V. Agrawal, bbox: 2D/3D bounding box library for Computer Vision. .
11. J. Huang et al., “Speed/accuracy trade-offs for modern convolutional object detectors,” arXiv:1611.10012 [cs], Nov. 2016.
12. “YOLO: Real-Time Object Detection.” [Online]. Available: https://pjreddie.com/darknet/yolo/. [Accessed: 24-Mar-2019].
13. R. Rothe, M. Guillaumin, and L. Van Gool, “Non-maximum Suppression for Object Detection by Passing Messages Between Windows,” in Computer Vision -- ACCV 2014, 2015, pp. 290–306.
14. D. H. Hubel and T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” J. Physiol. (Lond.), vol. 195, no. 1, pp. 215–243, Mar. 1968.
15. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
16. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, and Y. Bengio, “Dropout: A simple way to prevent neural networks from overfitting,” presented at the The Journal of Machine Learning Research.
17. J. Aarshay, “Deep Learning for Computer Vision – Introduction to Convolution Neural Networks,” 04-Apr-2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/04/deep-learning-computer-vision-introduction-convolution-neural-networks/.