ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

Sheep Ages Recognition Based on Teeth Images

Hadi Yahia Albrkaty
Master Research Project Report
Umm Al-Qura University (UQU)

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

TABLE OF CONTENT

Contents **Page No.**

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

LIST OF TABLES

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

LIST OF FIGURES

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

ACKNOWLEDGEMENTS

This work was supported to fulfill the requirements for obtaining a degree Master of Science in Computer Science and Engineering (Artificial Intelligence) at Umm Al-Qura Uni- versity, my special thanks to everyone who cooperated with me, including the owners of sheep, which are sheep of AL-Shamkhat by Obaid albarakati, Al-Tha'labi and Khaled Al-Qahtani's , and a special greeting I extend to my father and mother for their moral and psychological support and to all anonymous reviewers for their suggestions

ABSTRACT

The health of sheep's teeth affects the abundance of meat and their good health through their healthy teeth, as it may cause their teeth to erode or break due to the presence of lean sheep. Also, by looking at the teeth of sheep, we can categorize them according to their ages to deal with each type as needed. Knowing the sheep age from their teeth is a pure sheep owners and shepherds' skill.

The spread of cell phones presents an opportunity for any people to benefit from many applications that make strange and difficult domains familiar to the public. Designing and implementing a sheep ages recognition system would significantly affect the speed and quality work of many buyers, sellers and interested people.

The proposed project aims at addressing the Sheep ages recognition problem. A number of efficient deep learning architectures will be used, in order to select the best one that ensure the trade-off between optimizing the classification performance and model size. Moreover, a real dataset will be collected for 3 different sheep ages and an appropriate performance metrics will be used to evaluate the different proposed models. Besides, pre-processing and data augmentation techniques will be investigated to overcome the collected data.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER ONE
INTRODUCTION

The history of the domestic sheep goes back to between 11,000 and 9,000 BC, and Sheep are among the first animals to have been domesticated by humans. These sheep were primarily raised for meat, milk, and skins. Woolly sheep began to be developed around 6000 BC. They were then imported to Africa and Europe via trading.

The length of a sheep's productive lifetime tends to be much less. This is because an ewe's productivity usually peaks between 3 and 6 years of age and begins to decline after the age of 7. As a result, most ewes are removed from a flock before they would reach their natural life expectancy. It is also necessary to get rid of older ewes in order to make room for younger ones. The younger animals are usually genetically superior to the older ones. In harsh environments (e.g., where forage is sparse), ewes are usually culled at a younger age because once their teeth start to wear and break down, it becomes more difficult for them to maintain their body condition and consume enough forage to feed their babies. It is possible for an ewe to be productive past 10 years of age, if she is well-fed and managed and stays healthy and sound.

- *Sheep Breeds*

It is difficult to know how many breeds of sheep there are in the world, as only developed countries usually maintain breed registries. However, it is believed that there are more breeds of sheep than breeds of any other livestock species, with the exception of poultry. Worldwide, it is estimated that there are more than 1000 distinct sheep breeds. There are more than 60 breeds in the United States alone. Sheep come in all different sizes, shapes, and colors, and there are many ways to classify sheep: according to their primary purpose (meat, milk, or wool), the type of coat they have or fibers they grow (fine, medium, long or carpet wool; or hair), the color of their faces (black, white, red, or muddled), and/or by specific physical or production characteristics.

- *Basic Sheep Care*

Pet sheep have most of the same basic needs as sheep being raised for production. They need shelter and protection from predators. They need proper nutrition and health care. They need their hooves trimmed. Wooled sheep need to be sheared annually in a timely fashion, preferably by a trained professional

- *Meat*

The most important product we get from sheep is meat. Meat is an important component of our diets, and lamb and mutton supply us with many of the vital vitamins and proteins we need for healthy living. Lamb is the meat (flesh) from a sheep that is less than one year old. Mutton is the meat from a sheep that is over one year of age. The terms yearling mutton are applied to the meat from a sheep that is between one and two years of age. While sheep meat only accounts for 6 percent of the world's meat consumption, it is the principal meat in regions of North Africa, the Middle East, India, and parts of Europe. The European Union is the world's largest lamb consumer and number one importer of lamb, whereas 99 percent of the lamb imported originates from Australia and New Zealand.

- *Aging Sheep*

The teeth of a sheep are divided into two distinct sections, namely, eight permanent incisors in the lower front jaw and twenty-four molars, the latter being divided into six on each side of the upper and lower jaw.

Sheep have no teeth in the front part of the upper jaw which consists of a dense, hard, fibrous pad. When born, the lamb usually has no teeth. Within a week after birth, the milk teeth or temporary teeth appear in the front lower jaw and by the time the lamb is two months old these, eight in all, have erupted. These temporary teeth are replaced by permanent incisors, which appear in pairs, commencing with the two central teeth, followed by one on either side at intervals, until the eight temporary teeth have been replaced. During the period the teeth are growing, sheep are referred to by the number of permanent incisors present, such as two-tooth, four-tooth, six-tooth, eight-tooth or full mouth.

Only a rough estimate of a sheep’s age can be made by looking at its teeth. When estimating the age, it is important to bear in mind whether the breed is early or late maturing. The condition of the teeth will vary according to the type of feed and country grazed on. On long, soft feed the teeth will grow long from lack of wear, but remain in good condition. On short feed, where close grazing is necessary, particularly if the soil is sandy or gravelly, the teeth will wear down. After the eight permanent incisors have appeared, the next stage is known as ‘broken mouth’. This is a progressive deterioration, the rate depending on the conditions under which the sheep was grown. Estimation of age at this stage is very difficult. The teeth gradually become longer with wide spaces, eventually falling out, or they may wear down, become loose and fall out. After the teeth have fallen out the sheep is known as a ‘gummy’.

To summarize, the approximate age of a sheep can be determined by examining their upper incisor teeth. At birth, lambs have eight baby (or milk) teeth or temporary incisors arranged on their lower jaw. They don't have any teeth on their top jaw, only a dental pad.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

At approximately one year of age, the central pair of baby teeth is replaced by a pair of permanent incisors. At age 2, the second pair is replaced by permanent incisors. At 3 and 4 years, the third and fourth pairs of baby teeth are replaced.

At approximately four years of age, a sheep has a full mouth of teeth. As it ages past four, the incisor teeth will start to spread, wear, and eventually break. When a ewe has lost some of her teeth, she's called a "broken mouth" ewe. When she's lost all her teeth, she's called a "gummer."

A sheep with no incisor teeth can still survive because it uses mostly its molars for chewing feed. However, it will have a harder time grazing, especially short vegetation.

Figure 1 shows the standard teeth variation that expresses the sheep ages. The purpose of this project is to design and implement a sheep ages recognition system. The proposed system uses deep learning algorithms to learn the discriminant visual features that characterize teeth forms and shapes from a set of images presented as a data set for the first time in the field of artificial intelligence (according to my knowledge) specialized in sheep's teeth to identify their ages and determine the ages of sheep through their teeth, and build a classification model able to well generalize on unseen teeth images.

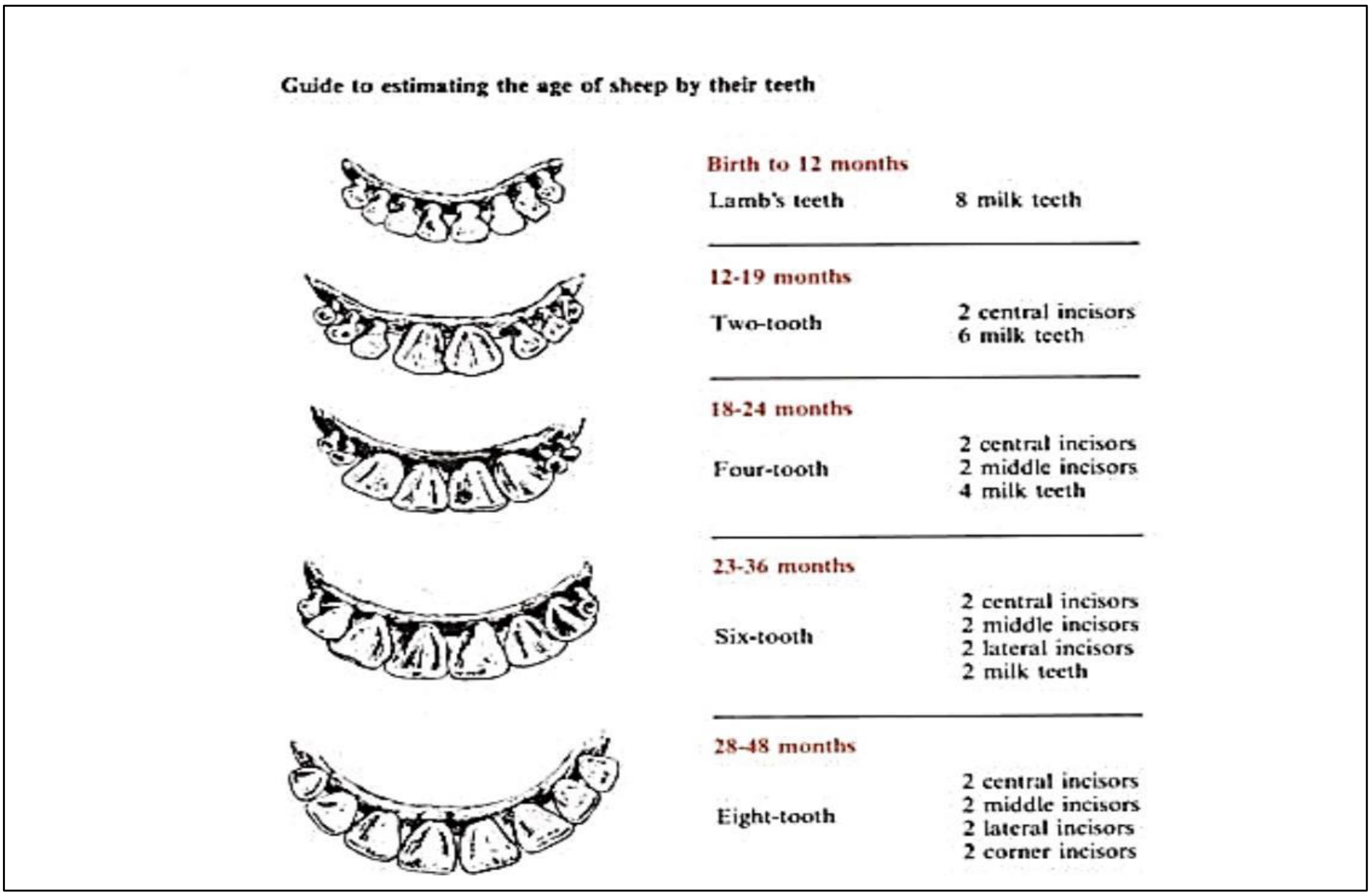

Fig 1 Teeth Shape According to Age [1].

➢ *Motivation*

Hajj is in the twelfth month. Performing the Hajj requires slaughtering the sacrificial animal, which is considered a sacrifice offered to complete the religious ritual. This sacrifice is made of sheep for pilgrims and non-pilgrims at certain times. These sheep require a certain age to be discovered from their teeth by opening the sheep's mouth and looking directly at their teeth, which determines the requirements of the sacrifice and whether it is acceptable or not. This is from stimulating the implementation of the project and automating it by artificial intelligence for automatic identification and work on it. In addition, the examination of sheep's teeth is useful in discovering sheep's dental problems, as the safety of the teeth determines that these sheep enjoy good health because they are able to chew the feed well, unlike the sheep that cannot eat because of the erosion and breakage of the teeth. There are scientific studies that suggested the reasons for the weakness and thinness of some sheep due to tooth decay. In this research project, I will do a lot of effort and continuous work in order to make a model that predicts through neural networks and deep learning the ages of sheep through their teeth and compare a group of models to choose the best model that does this with the highest accuracy and least errors. We have divided and categorized the group of pictures into 3 classes, the first has 2 large incisors and has one year old, the second has 4 large incisors and is two years old, and the third has 6 incisors (3 incisors on each side) and is 3 years old (As the figure 1 shows).

The data was trained on five models in deep learning by the PyTorch platform, and the best result of the best model was evaluated on test data set. The evaluation is done by measuring f1-score, accuracy, recall, and precision, and confusion matrix.

- *Problem Statement*

The proposed system is an image-based solution. Machine learning can give a solution to this system. The performance of such systems depends on the quality and the low-level features chosen to encode the teeth images. However, the emergence of convolutional neural network (CNN) and their efficiently to extract features maps from images across researches promoted its use to address sheep ages recognition problem. Most deep learning models are CNN-architecture based and their main limitation is their high computational cost due to the architecture complexity.

- *Goals and Objectives*

This project aims to investigate the recognition approaches based on images and implement a CNN-based architecture models. The best model will be deployed initially in a web application and finally in a mobile application.

- *Solution*

In this project, we propose an image-based sheep ages recognition system. A group of well-known efficient CNN-based deep learning models are exploited to increase the overall teeth ages recognition performance. Unprecedented and Real datasets will be considered to evaluate the performance of the proposed system.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER TWO
LITERATURE REVIEW

➢ *Background*

Sheep's teeth can reveal their age and health. Teeth shape, sharpness, width and length develop and change with age. Automatic recognition based on images is a popular idea in the scientific community and industry, while the idea of recognizing the ages of sheep based on images is a new idea as far as we know. In this chapter, we present an overview on the topics relevant to automatic recognition from images.

➢ *Image-based Recognition*

Recognition systems contain four principal steps, where each single step aims to achieve a particular task. Those five stages are: acquisition of images, preprocessing the images, then extraction the features from those images, and lastly classification process.

The system is fed with an image that includes an open mouth of a sheep. The system is facing some challenges including highly variant image background and noisy images (e.g., images taken with incomplete teeth). Obviously, segmenting images, by just taking the interested area-mouth space- reduces the computational cost [2]. Some deep learning systems are fed with the entire images, since they are able to detect the interested area implicitly.

The system performance depends on the quality of features extracted. In machine learning systems, the features are handcraft, i.e., extracted manually, and this require knowledge domain. The efficiency of deep learning systems comes from the fact they extract the features implicitly. As for classification step is the process of using the features extracted to distinct the different classes.

➢ *Deep Learning*

Deep learning is considered as a sub-group of machine learning [3]. It has provided efficient solutions for image-based recognition problems. Deep learning consists of a several neural network layers stacked one after another. The more layers it has, the deeper it is. A very simple neural network is composed of one or more hidden layer. A hidden layer is just a number of neurons. Figure 2 shows such a network.

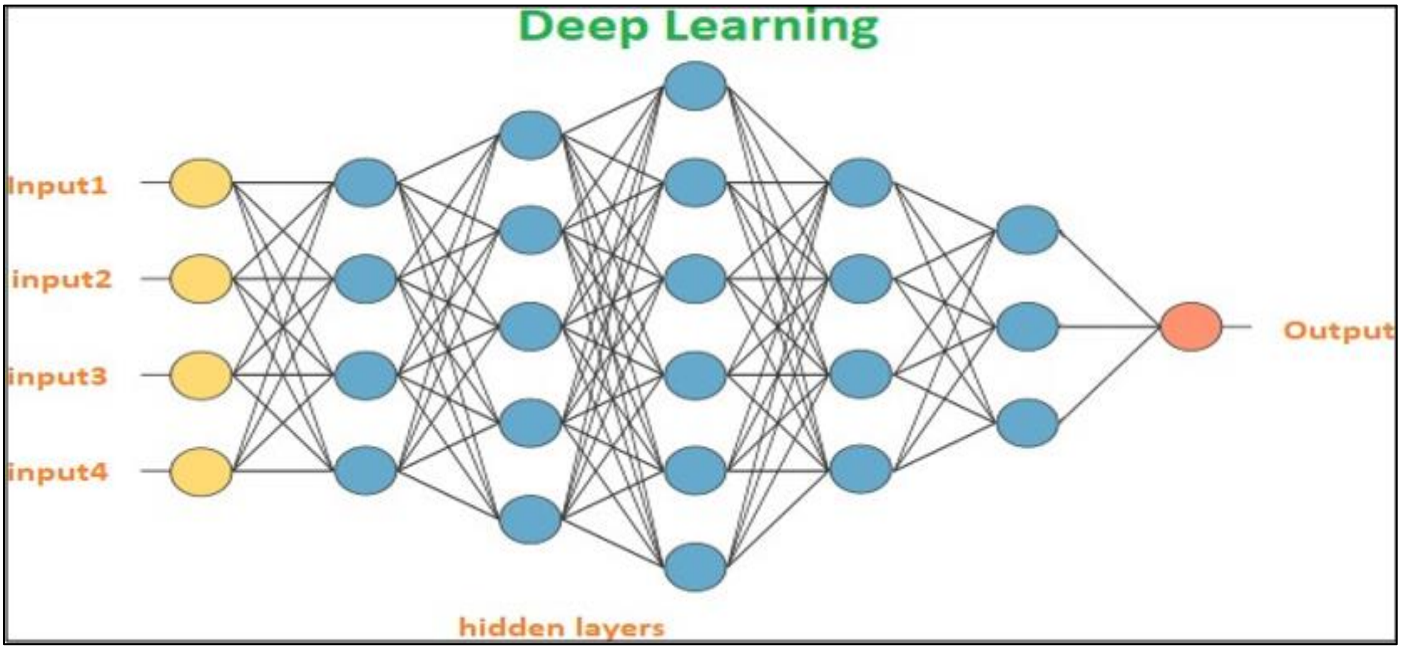

Fig 2 Architecture of Neural Network

Deep learning algorithms mainly differ in the type of layers that compose them. Convolutional neural network (CNN) is considered as a type of deep networks that is generally applied in computer vision. Typically, a CNN consists of stacked layers that containing the convolutional, pooling and fully connected layers (hidden layers) as illustrated in Figure 3.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

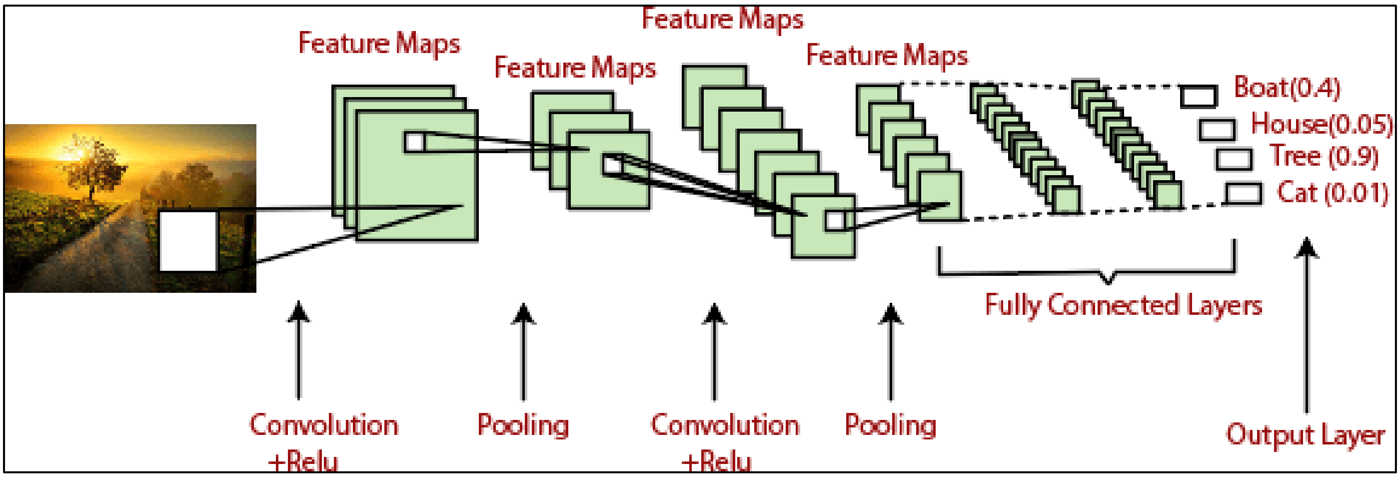

Fig 3 Architecture of CNN.

The convolution layer is the process of computing the convolution between what is called a filter and a part of image. This process aims at detecting relationship between image pixels. The result of this operation is the feature image. In fact, we use mor than one filter in a convolution layer. The result then is a feature map (feature images stacked one after another). The pooling layer aims at reducing the dimensionality by keeping the most relevant part in feature image. The fully connected layer consists of hidden layer. It is used for classification since we can easily change the number of the output classes [4].

There are many tasks ich benefit of CNN-based deep learning. Face recognition, emotion analysis, super resolution, and multimedia structures are ones of these tasks [5].

During the training, deep leaning models predict outputs for the images. A loss value is then computed to evaluate how these outputs are close the image targets/classes. The model learns if the loss is reducing. Hence, the process of training a deep learning model is the process of reducing the loss cost/function. Reducing the loss function is typically done by an optimizing an algorithm for stochastic gradient descent [6]. The process of optimization leads to the process of updating the weights and bias layers of the deep learning model. Changing these weights leads to minimize the loss function [7]. There are several algorithms used for optimization. Adam algorithm is the one used in this work. It is a gradient-based optimization methods and combines the advantages of Adaptive Gradient Algorithm -Adagrad- and RMSprop; an extension of Adagrad algorithm [6].

➢ *Literature Review*

In order to enhance the performance of identification sheep systems, authors in [8], provide a dataset consisting of 416 color images for different features of sheep in different postures. Images were collected fifty-two sheep at a range of year from three months to six years. For each sheep, two images were captured for both sides of the body, two images for both sides of the face, one image from the top view, one image for the hip and one image for the teeth. The collected images cover different illumination, quality levels and angle of rotation. The allocated data set can be used to test sheep identification, weigh estimation, and age detection algorithms. Such algorithms are crucial for disease management, animal assessment and ownership.

Project conducted in [9] provides a pilot evaluation on the utility of deep learning in accurately predicting performance outcomes from sheep images, biomarkers and on-animal sensor output. The project also undertook a global review of new bio-markers and bio-sensors that may have applicability within the sheep industry.

Limb movements and activities can directly reflect the adaptability of sheep to the breeding environment and conditions, and provide a richer scientific and technical experience for sheep breeding. The sheep target detection is an important prerequisite for grasping the movement behavior of sheep. Authors in [10], uses the Faster-RCNN neural network model based on the Soft-NMS algorithm to realize the real-time detection and positioning of sheep under complex breeding conditions, and improves the accuracy of recognition while ensuring the detection speed.

Authors in [17] used the Faster-RCNN neural network model based on the Soft-NMS algorithm to realize the real-time detection and positioning of sheep under complex breeding conditions. They improved the accuracy of recognition while ensuring the detection speed. Experiments show that their proposed detection model can detect sheep with 95.32% accuracy and mark the location of the target in real time, which provides an effective data foundation for sheep behaviour research and helps promote the development of high-tech animal husbandry effect.

To the best of our knowledge, there are no precedents works studying sheep age recognition based on teeth images. In the next section, the proposed approach for sheep age image-based recognition.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER THREE
PROPOSED APPROACH

➢ *Introduction*

We propose sheep age image-based recognition system. The proposed system is based on CNN deep learning. It is intended to convert automatically the open mouth/teeth images into the corresponding Age. As shown in Figure 4, the sheep age recognition problem is tackled as a supervised deep learning. Addional to a baseline CNN model, we propose a comparative study of the five most efficient deep learning models in the literature.

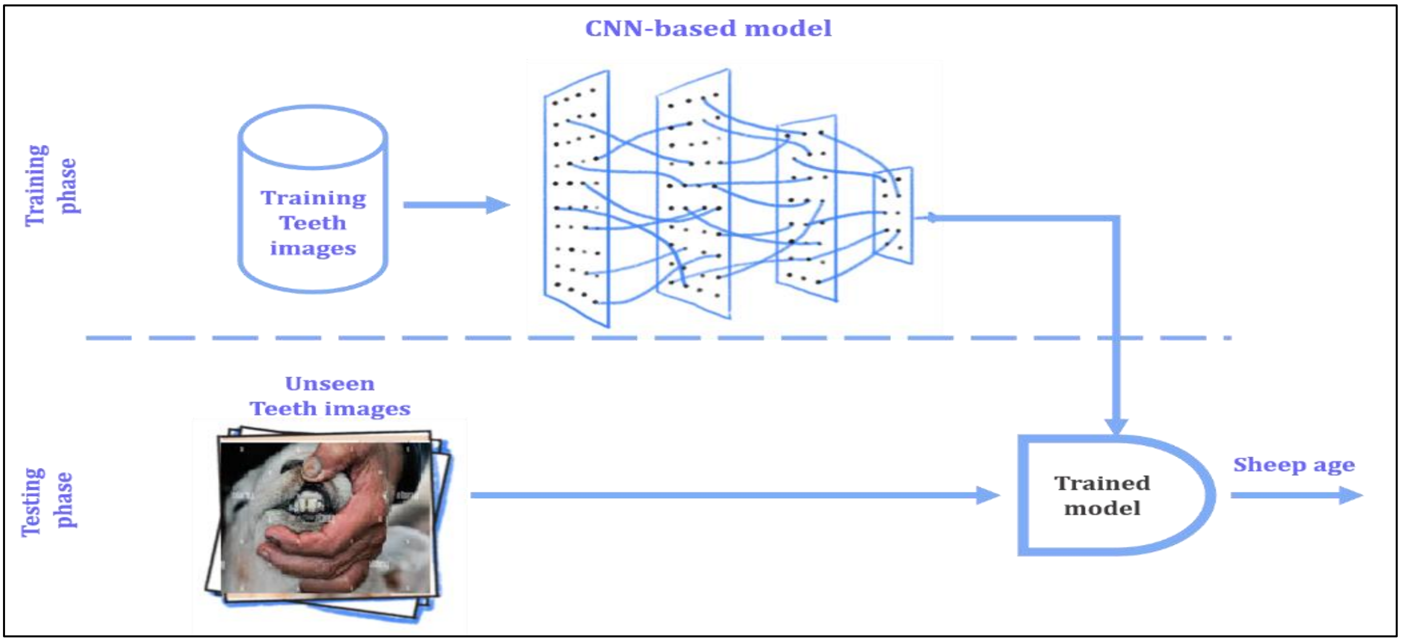

Fig 4 Proposed Approach of Sheep Age Recognition.

➢ *Proposed Deep Learning Models*

In this project, we plan to apply 6 CNN-based deep learning models. A bassline model and another five efficient models in image classification domain. CNN Models differ principally by the number of layers, size kernel of the filter, how the layers are organized and more. In this section, we will give a compact view of the proposed models.

- *VGG-16 Architecture*

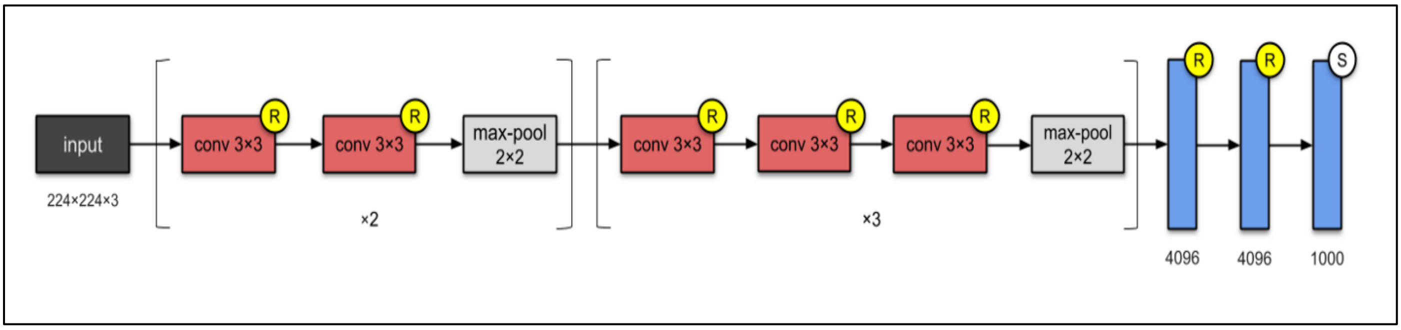

Fig 5 VGG-16 Architecture [11].

Visual Geometry Group (VGG) invented the VGG-16 which has 13 convolutional layers, 2 max-pooling, and 3 fully-connected layers, carrying with them the ReLU activation function. It uses filter with 3x3 kernel size. The number 16 refers to the number of trainable layers. Max-pooling layer is not out of them, since it has not a weights to train. This model is the first deeper model by stacking uniform convolutions. VGG-16 has 138 M parameters and it is considered a large model.

- *Inception-v3 Architecture*

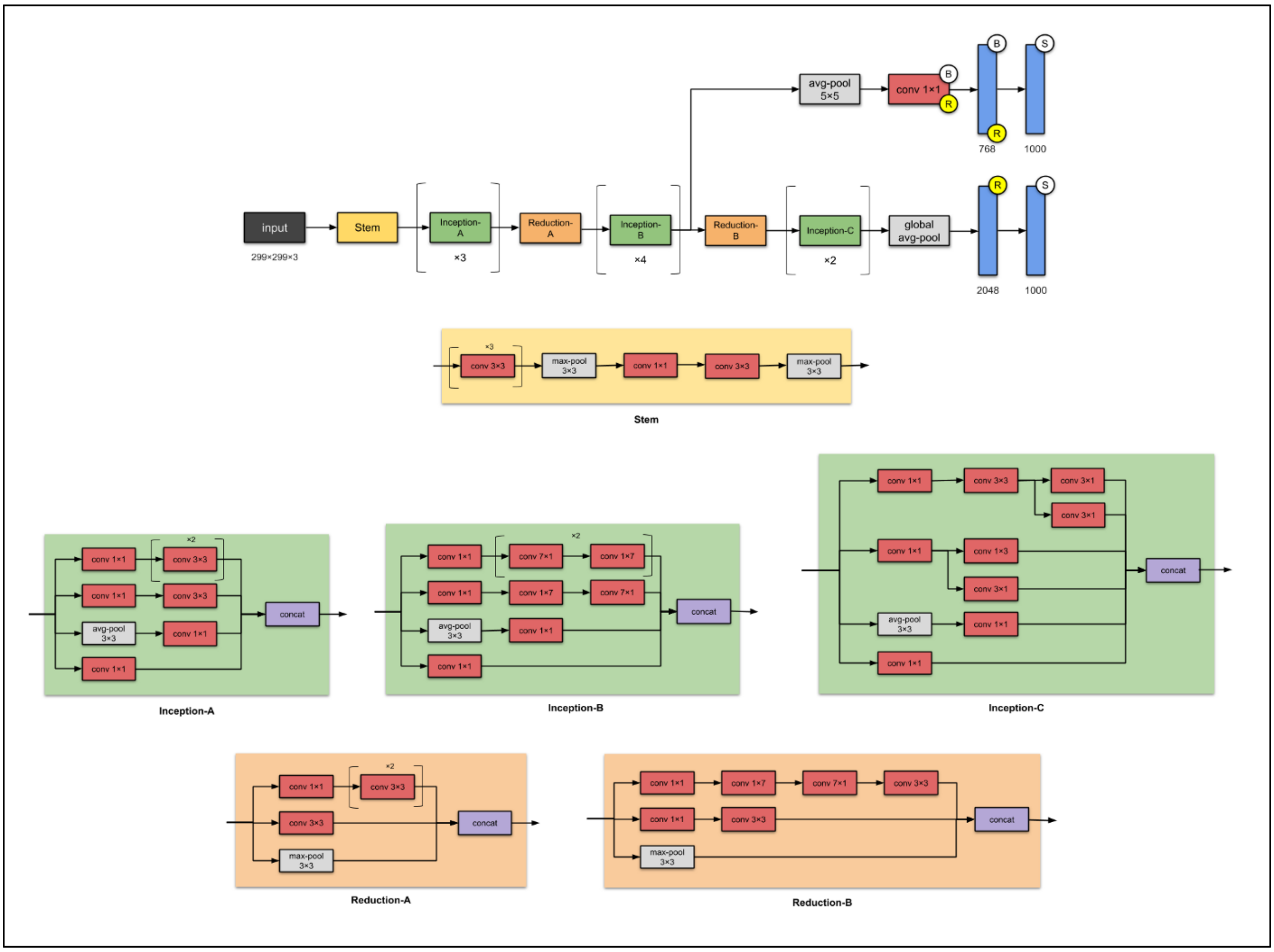

Fig 6 Inception-v3 Architecture [12].

Figure 6 shows towers of convolutions with series of two 3×3 convolution operations instead of 5×5 and 7x7 convolution, to reduce computational complexity. The 1×1 convolutions are used for dimensionality reduction to reduce memory requirement. two auxiliary classifiers introduced to encourage discrimination in the lower stages of the classifier, to increase the gradient signal that gets propagated back, and to provide additional regularization. The branches that are connected to the auxiliary classifier are discarded at prediction time. The new with this model is instead of stacking convolutional layers, we stack blocks, within which are convolutional layers.

- *ResNet Architecture*

So far we have seen that models are getting deeper by adding layers and blocks. And this is good at the level of extracting features. But we will face a vanishing gradient problem; the weights of the first layers stop updating and hence the model stops updating and the accuracy will be saturated. This is because when the network is too deep, the gradients from where the loss function is calculated tend to be zero after several applications of the chain rule.

ResNet solve this problem by introducing the residual connection, which allows the gradients flow directly through the skip connections backwards from later layers to initial ones. Figure 7 depicts this concept. 50 refers to the number of layers in this version.

- *ResNext Architecture*

ResNext is a marriage of VGG, ResNet, and Inception, composed via repeating a block as in VGG that aggregates a set of transforms like Inception while bearing residual connections like in ResNet. The novel in ReNext is rather than performing convolutions over the full input feature map, the block's input is projected into a series of lower (channel) dimensional representations of which we separately apply a few convolutional filters before merging the results. Figure 8 and 9 illustrates theses new features.

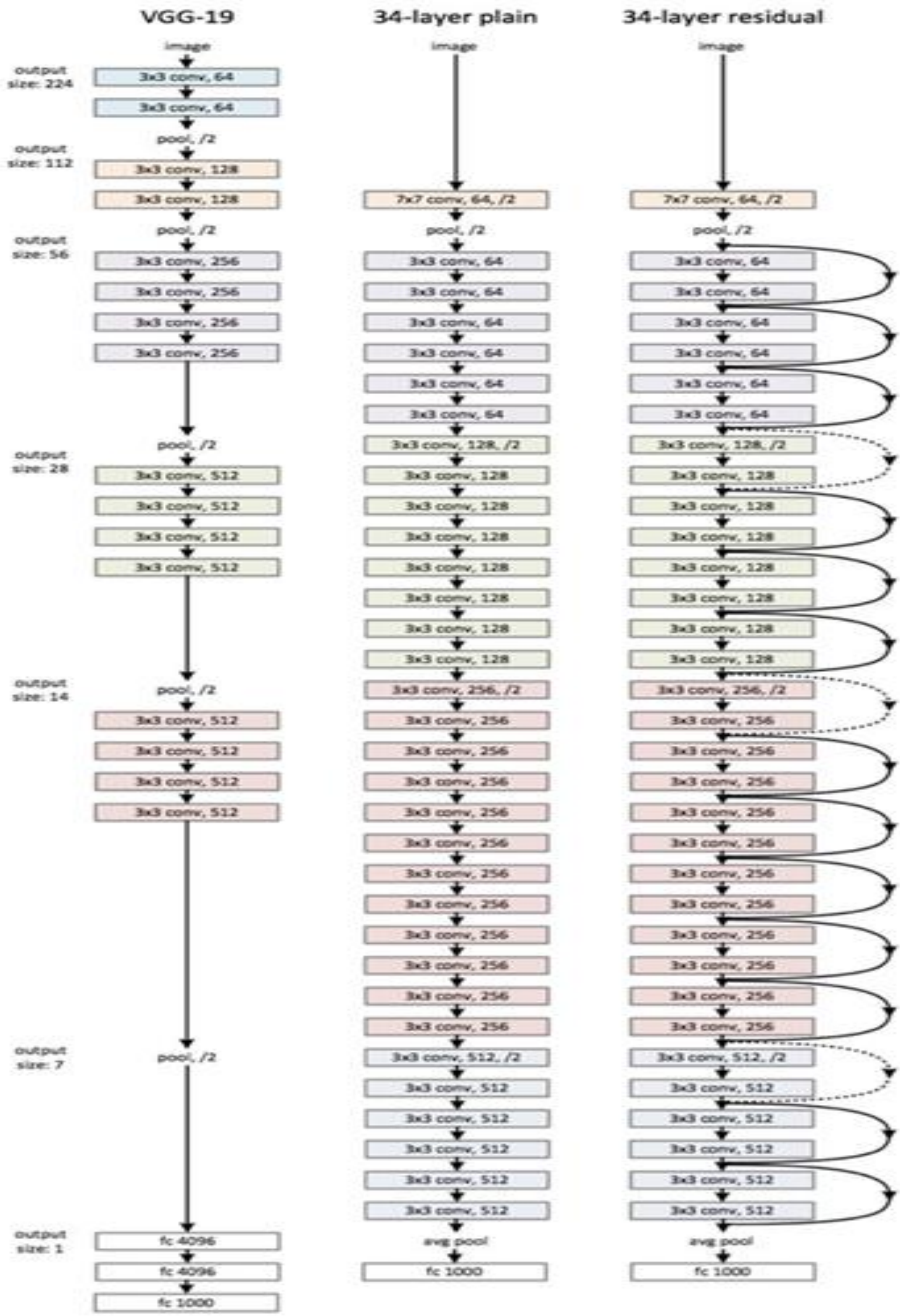

Fig 7 Residual Connection-ResNet Architecture [13].

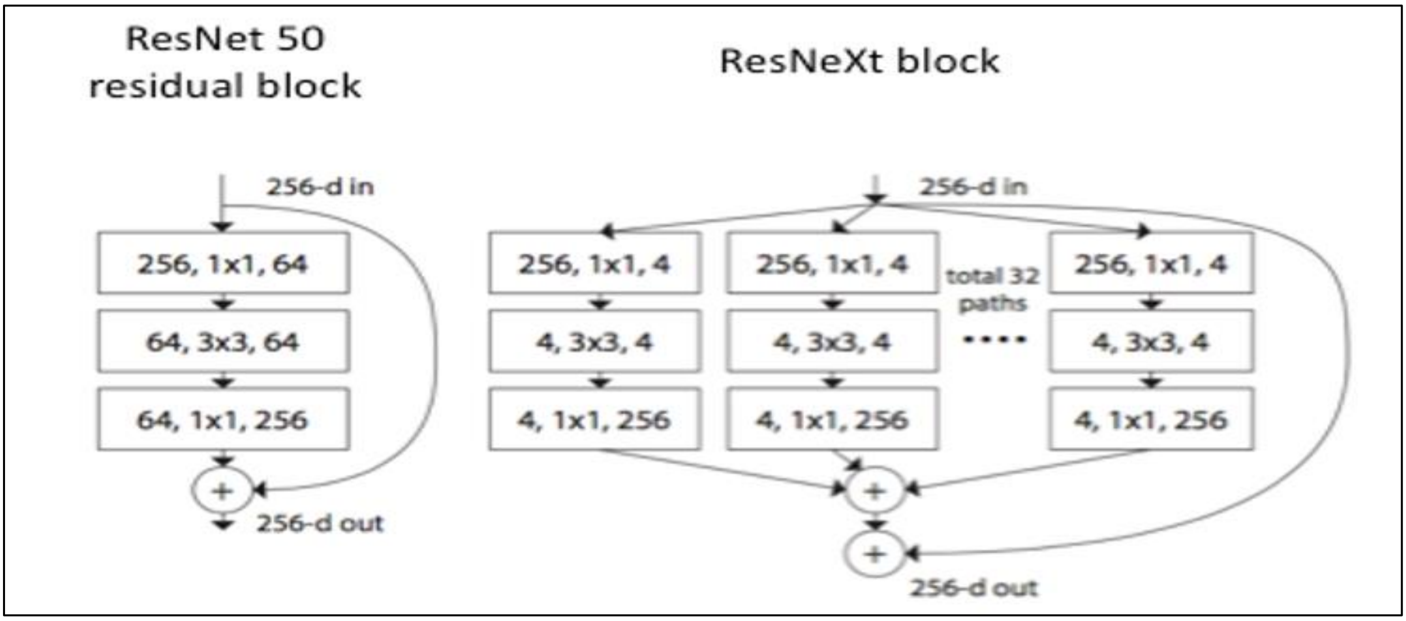

Fig 8 Block in ResNext Architecture

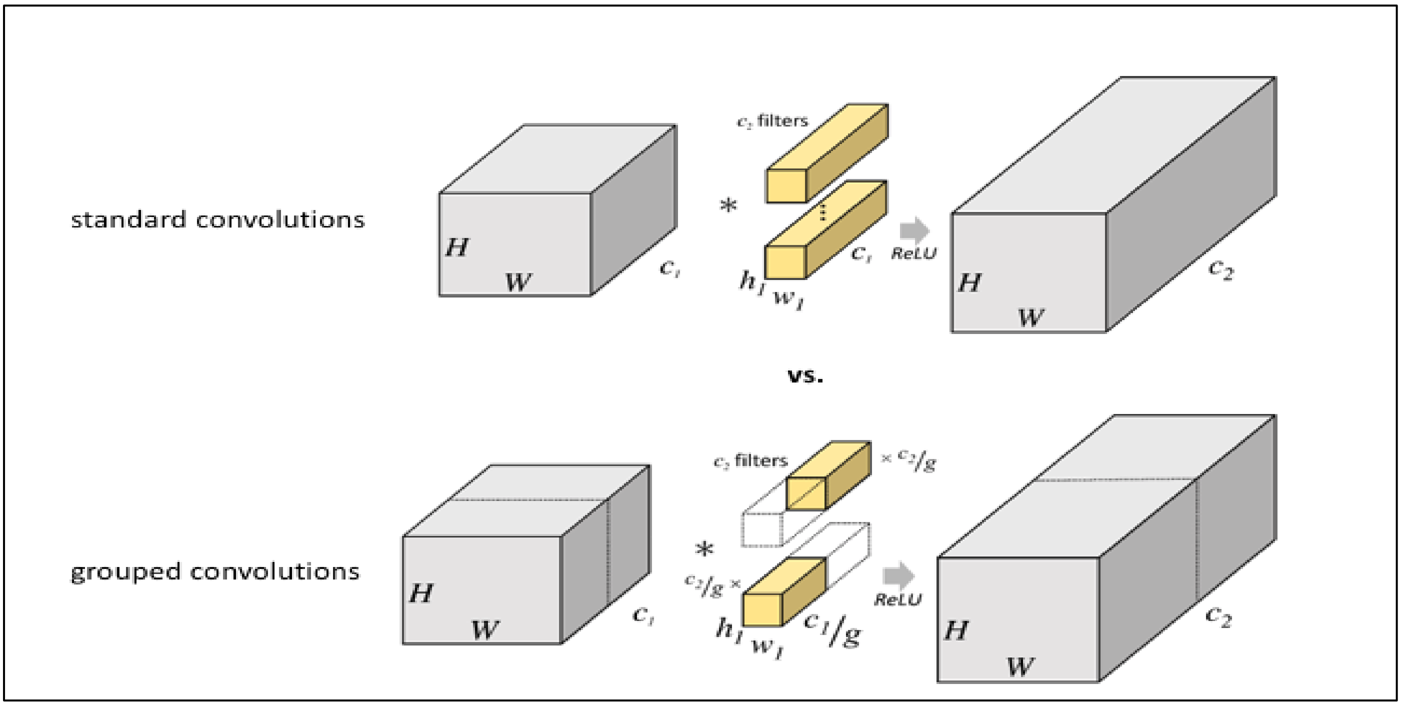

Fig 9 Cconvolution in ResNext Architecture

- *EfficientNet Architecture*

EfficientNet [15] uses a technique called compound coefficient to scale up models in a simple but effective manner (Figure 10). Instead of randomly scaling up width, depth or resolution, compound scaling uniformly scales each dimension with a certain fixed set of scaling coefficients. Using the scaling method and AutoML, the authors of efficient developed seven models of various dimensions, which surpassed the state-of-the-art accuracy of most convolutional neural networks, and with much better efficiency. EfficientNet is based on the baseline network developed by the neural architecture search using the AutoML MNAS framework. The network is fine-tuned for obtaining maximum accuracy but is also penalized if the network is very computationally heavy. It is also penalized for slow inference time when the network takes a lot of time to make predictions. The architecture uses a mobile inverted bottleneck convolution similar to MobileNet V2 but is much larger due to the increase in FLOPS. This baseline model is scaled up to obtain the family of EfficientNets. Figure 11 shows the performance of EfficientNet compared to other network architectures. The biggest EfficientNet model EfficientNet B7 obtained state-of-the-art performance on the ImageNet and the CIFAR-100 datasets. It obtained around 84.4% top-1/and 97.3% top-5 accuracy on ImageNet. Also, the model size was 8.4 times smaller and 6.1 times faster than the previous best CNN model. It obtained 91.7% accuracy on the CIFAR-100 data set and 98.8% accuracy on the Flowers dataset.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

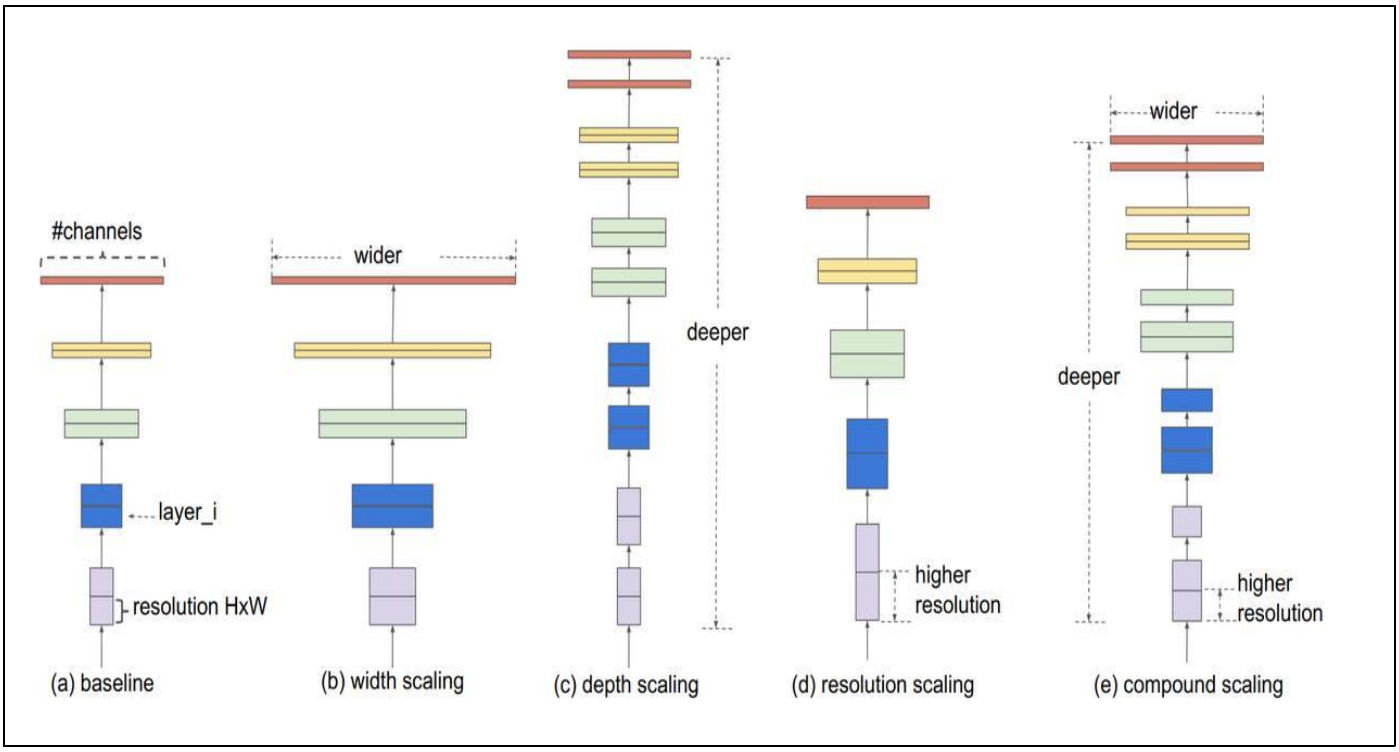

Fig 10 Scaling Method [15].

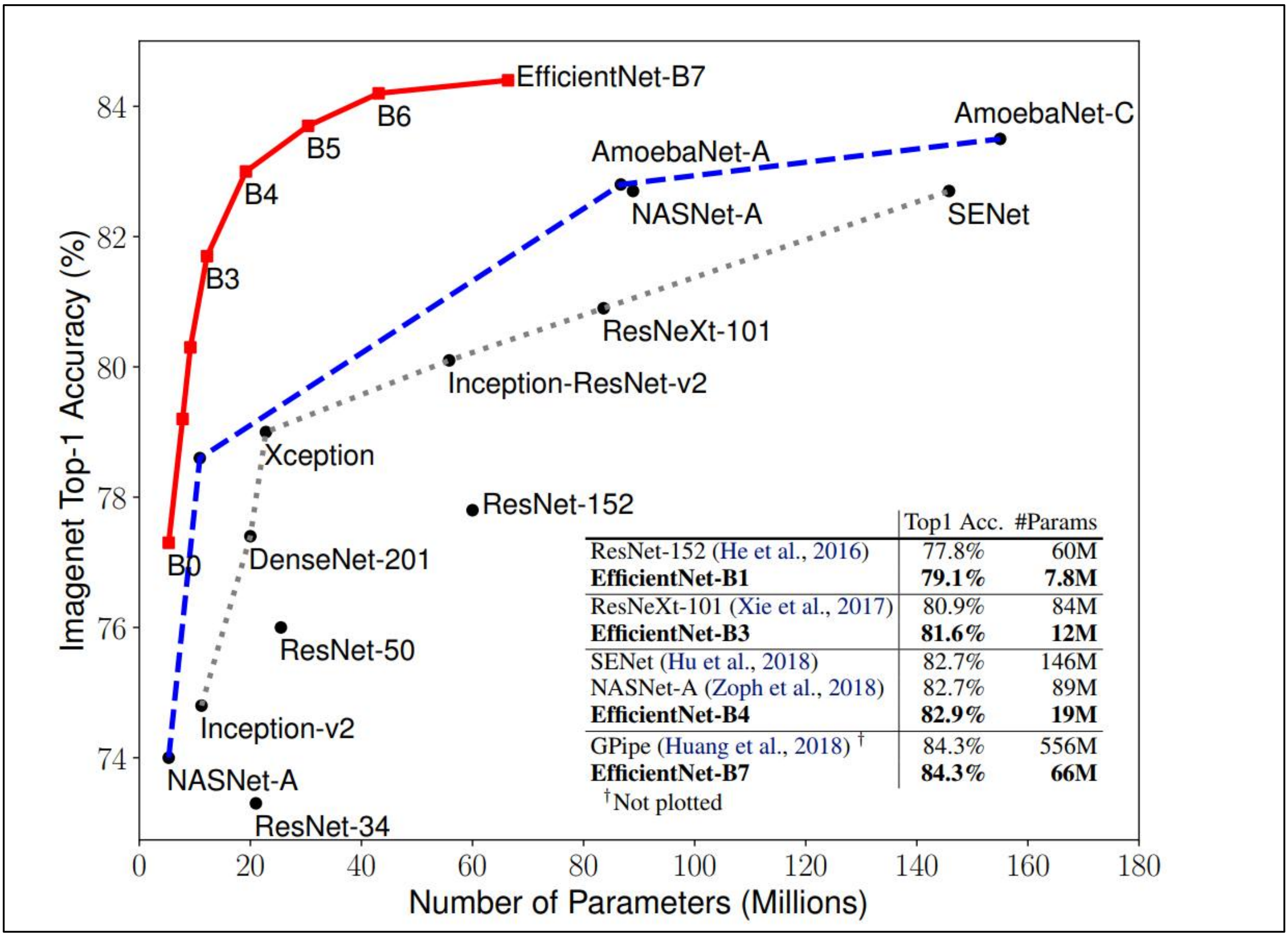

Fig 11 Performance of EfficientNet Compared to other Network Architectures [15].

Volume 9, Issue 9, September– 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER FOUR
EXPERIMENT SETTINGS

In this section, we present the sheep teeth dataset that will be used in this project. Furthermore, we explain the framework in which of the proposed models will be trained. After that, we describe the performance measures that will be used to evaluate the system performance.

- *Dataset Description*

In this project, we collected real teeth images to be used in the experiments. They distributed over 3 classes: 1 year sheep, 2 years sheep and 3 years sheep. Figure12 show some images from the dataset, while Figure 13 shows some statistics on it.

Fig 12 Sheep in Different Ages

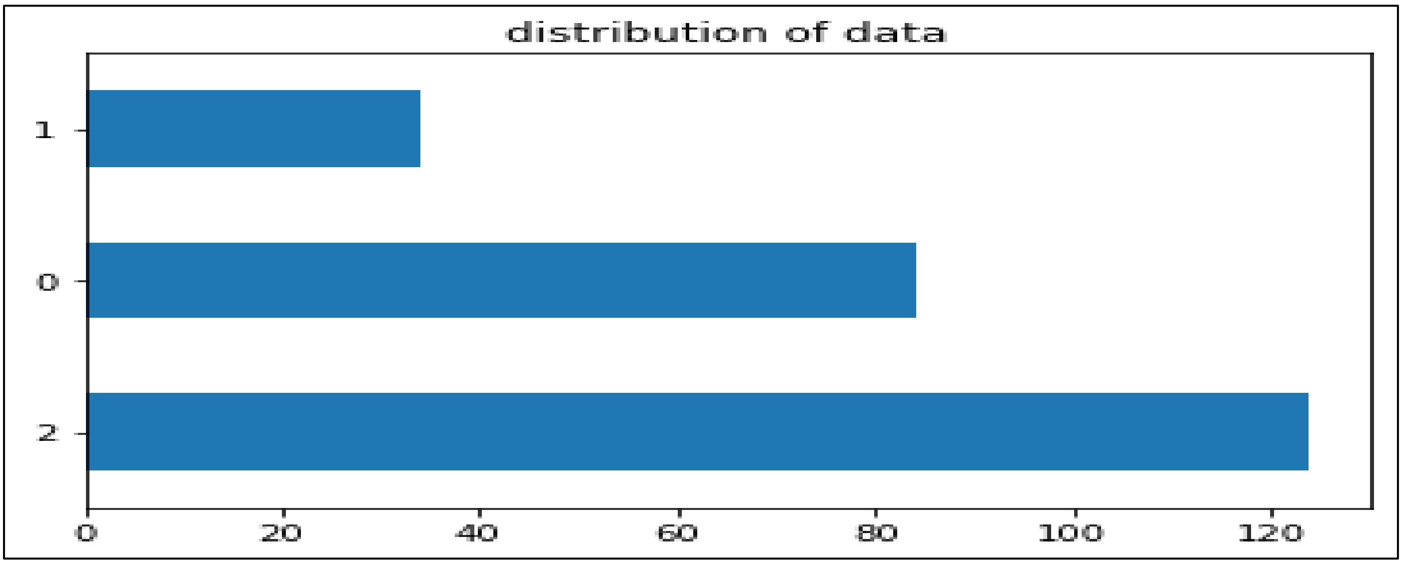

Fig 13 Data Distribution.

- Class 0: 1 year sheep, class1: 2 years sheep and class 2: 3 years sheep.

➢ *Challenges in Collecting Data*

The dataset is small and imbalanced. There are a number of challenges that we faced, which were the reason for the lack of data set, including that taking sheep's teeth pictures requires the help of a group of people to avoid shaking the sheep's head (at least 2 people, one catches the sheep and opens its mouth, and another takes the picture). Because it is very important that the images be clear and of high quality.

Furthermore, the collecting process forces the photography to be in the light to take good pictures. This be can painful since this project occurred during summer season where the weather in the region (the city of Makkah in the Kingdom of Saudi Arabia) is very hot during the day and the temperature often reaches about 50°C.

Another obstacle that caused less data to be collected was the embarrassment we were causing to the owner of the sheep, as while taking pictures, some of the sheep were escaping into the vast spaces.

➢ *Performance Measures*

In our project we considered four different performance measures. Namely: accuracy, measure precision, recall and F1-score. F1-score is more important than the accuracy in the case of imbalanced data. Below are the equations used to compute ach measure.

Let us know that *True positive* is the result when the model correctly predicts the positive class. *True negative* is the result when the model correctly predicts the negative class. *False positive* is the result when the model incorrectly predicts the positive class. *False negative* is the result when the model incorrectly predicts the negative class.

- *Accuracy*

Accuracy is the ratio of correctly classified ages over the number of all classification for an and can be computed as follows:

$$\text{Accuracy} = \frac{\text{True(Positive + Negative)}}{\text{True(Positive + Negative) + False(Positive + Negative)}} \tag{1}$$

- *Precision*

The precision is the ratio of the positive classifications results that are indeed positive and can be computed using the eqution (2).

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive+False positive}} \tag{2}$$

- *Recall is the Sensitivity of Classification and can be Computed as follows:*

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive+False Negative}} \tag{3}$$

- *F-1 Measure is the Tradeoff between Precision and Recall Measures and it is Computed by the Following Equation:*

$$\text{F} - 1\ \text{Measure} = \frac{\text{Precision} * \text{Recall}}{\text{Precision + Recall}} \tag{4}$$

➢ *Training Framework*

Unlike traditional machine learning, deep learning algorithms require big amount of data to generalize and perform well. Collecting a sufficient amount of data is hard and expensive. Thanks to transfer learning, we can build a relatively good model with small size data.

Transfer learning is when a given deep learning model trained on one task (using a big dataset) is reused on a second related task (on related small dataset). As our task is image classification, we should use one of those models that were pretrained on images. As the dataset used to train the first model is different to our data, we transfer the learning by doing the following:

The architecture of the model is composed of a backbone and a classifier. The backbone part is generally composed of many layer/blocks and aims to extract the features from images, while the classifier consists of at last one fully connected layer. It has as goal to uses the features obtained from the backbone and classify them. Hence, the process of transfer the learning from the pretrained model can be done as follows:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

- Freeze the parameters of the backbone (They will not be trained again on our data)
- Modify the classifier such that the last layer will output the number of classes contained in our dataset instead of the number of classes in the big data used to train the model at the beginning.

Deep learning models are composed of layers, they differ from each other by the type and number of layers and how they connected between them. In this work we are going to use one CNN-based deep learning model and 5 pretrained models, namely, EfficientNet B0 version, Inception_v3, Resnet50, VGG-16, ResNext.

The bassline model is 5 convolution layers followed by batch normalized layer and between two consecutive convolution layers, a max-pooling layer takes a place. Figure 14 show a detailed implementation.

```
[ ]  class Baseline(nn.Module):
         def __init__(self):
             super(Baseline, self).__init__()

             self.conv1 = nn.Conv2d(in_channels=3, out_channels=12, kernel_size=5, stride=1, padding=1)
             self.bn1 = nn.BatchNorm2d(12)
             self.conv2 = nn.Conv2d(in_channels=12, out_channels=12, kernel_size=5, stride=1, padding=1)
             self.bn2 = nn.BatchNorm2d(12)
             self.pool = nn.MaxPool2d(2,2)
             self.conv4 = nn.Conv2d(in_channels=12, out_channels=24, kernel_size=5, stride=1, padding=1)
             self.bn4 = nn.BatchNorm2d(24)
             self.conv5 = nn.Conv2d(in_channels=24, out_channels=24, kernel_size=5, stride=1, padding=1)
             self.bn5 = nn.BatchNorm2d(24)
             self.fc1 = nn.Linear(24*106*106, 3)
             self.relu = nn.ReLU()

         def forward(self, input):
             output = self.relu(self.bn1(self.conv1(input)))
             output = self.relu(self.bn2(self.conv2(output)))
             output = self.pool(output)
             output = self.relu(self.bn4(self.conv4(output)))
             output = self.relu(self.bn5(self.conv5(output)))
             output = output.view(-1, 24*106*106)
             output = self.fc1(output)

             return output
```

Fig 14 Baseline Model Pytorch-based Implementation.

CHAPTER FIVE
EXPERIMENTS, RESULTS AND DISCUSSION

➢ *Experimental Settings*

In order to evaluate the proposed models, we conducted two sets of experimental. In the first sets we applied the baseline and five pretrained models on our data without data augmentation. In the second sets of experiments, we applied the same models on the augmented data. The dataset described in Section 4.1 is split into three subsets: (1) A training set including 195 (80%) images, (2) A test set that includes 10% of the total number of images (25 images), and (3) A validation set that represents 10% of the total number of images (22 images).

Data augmentation is a technique applied either to increase the amount of data or to make data rich and avoid overfitting, by modifying slightly the original images.

- *The Type of Transformation we Studied are:*

✓ Resize, since the pretrained models require a specific input shape.
✓ RandomAffine((30, 50)), we get a transformed image with a random degree between 30 and 50.
✓ RandomAdjustSharpness(sharpness_factor=2): randomly adjusts the sharpness of the given image.
✓ AutoAugment(policy= transforms.AutoAugmentPolicy.IMAGENET), augments data based on a given auto-augmentation policy.
✓ RandomInvert(): randomly inverts the colors of the given image.
✓ RandomHorizontalFlip(p=1): flip images horizontally
✓ ColorJitter(0.2, 0.2, 0.2, 0.2): change the color of images
✓ RandomRotation(degrees=(-5,5)),
✓ RandomSolarize(threshold=192.0),
✓ RandomAdjustSharpness(sharpness_factor=2)

The transformations were chosen such that they will not change the context of image for our classification task, we tested each transformation and only the *ColorJitter* and *RandomHorizontalFlip* transformations don't degrade the performance.

As described above, our dataset is unbalanced, where the 2 years class is less represented in the data. Transformations helped us to create a second version of data that differs to the original by the direction and of its images, where we apply the *horizentallFlip* transformation. A third dataset is created by applying *colorJitter* transformation to minority class (2 years images) to augment the number of images of this class in the whole data. The test data in the two set of experiment is the same. It is unbalanced and it wasn't subject to any transformation.

In all experiments, Cross Entropy function is used as loss function and Adam algorithm was used to optimize it. The training phase started with a learning rate of 1-e4.

All experiments were conducted using Nvidia K-80 GPUs associated with 16 GBs of RAM. The model classes as well as the training, validation and the test procedures were implemented using *Pytorch* library [16].

➢ *Results Discussion*

Table 1 summarizes the results obtained by the evaluation of the models trained on the original data, while the Table 2 reports the results obtained when these models were trained on the augmented data. Obviously, the results reported on Tables are obtained using the testing data; the unseen data.

Table 1 Model Evaluation on Test Data (Models Trained on the Original Data)

Models	**Measures**			
Baseline	56.00	Precision(%)	Recall(%)	F1-score(%)
Vgg16	60.00	81.67	54.7	53.04
Inception	80.00	40.44	44.16	41.83
ResNet	80.00	85.78	68.95	71.54
ResNext	84.00	86.71	70.09	71.72
EfficientNet	84.00	89.58	70.37	74.00

Table 2 Model Evaluation on Test Data (Models Trained on Augmented Data)

Models	**Measures**			
	Accuracy (%)	**Precision(%)**	**Recall(%)**	**F1-score(%)**
Baseline	68.00	77.08	57.83	61.51
Vgg16	64.00	63.82	63.82	63.82
Inception-v3	84.00	79.96	78.92	79.30
ResNet	80.00	90.74	74.07	78.43
ResNext	92.00	89.10	93.73	90.71
EfficientNet	92.00	93.94	86.32	88.67

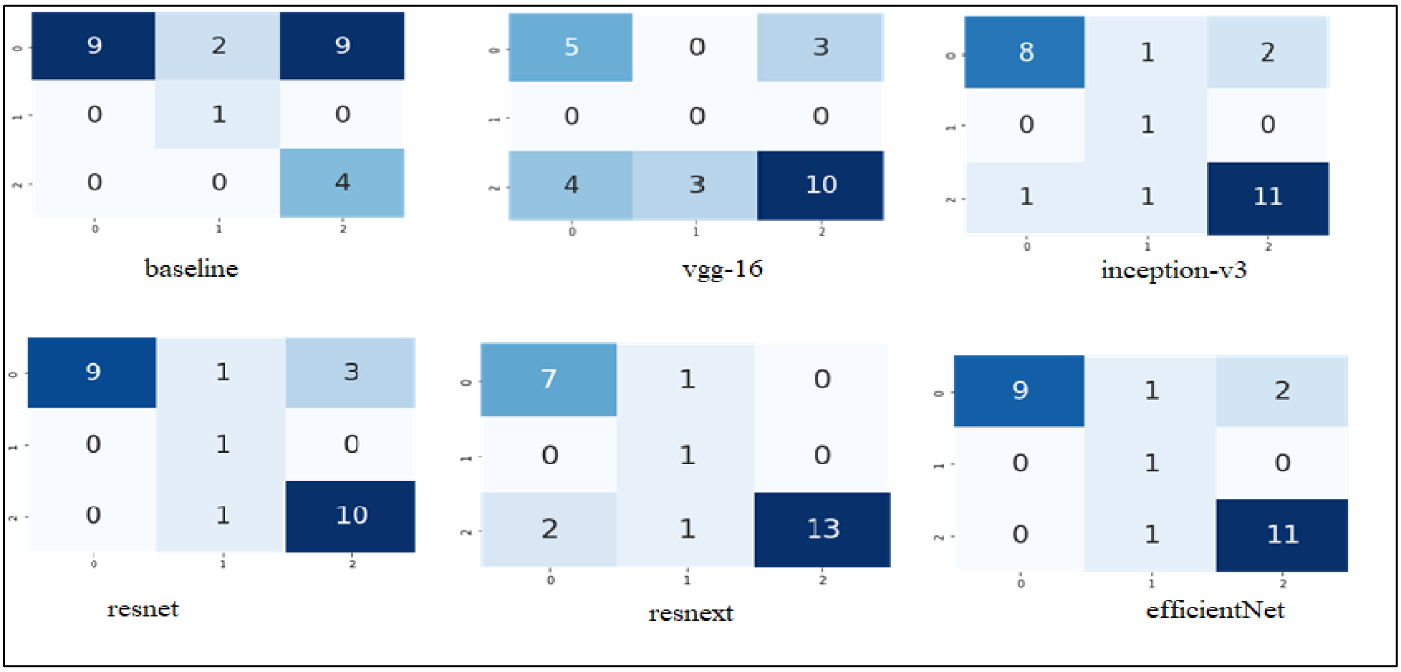

Fig 15 Confusion Matrix (Models Trained on Original Data)

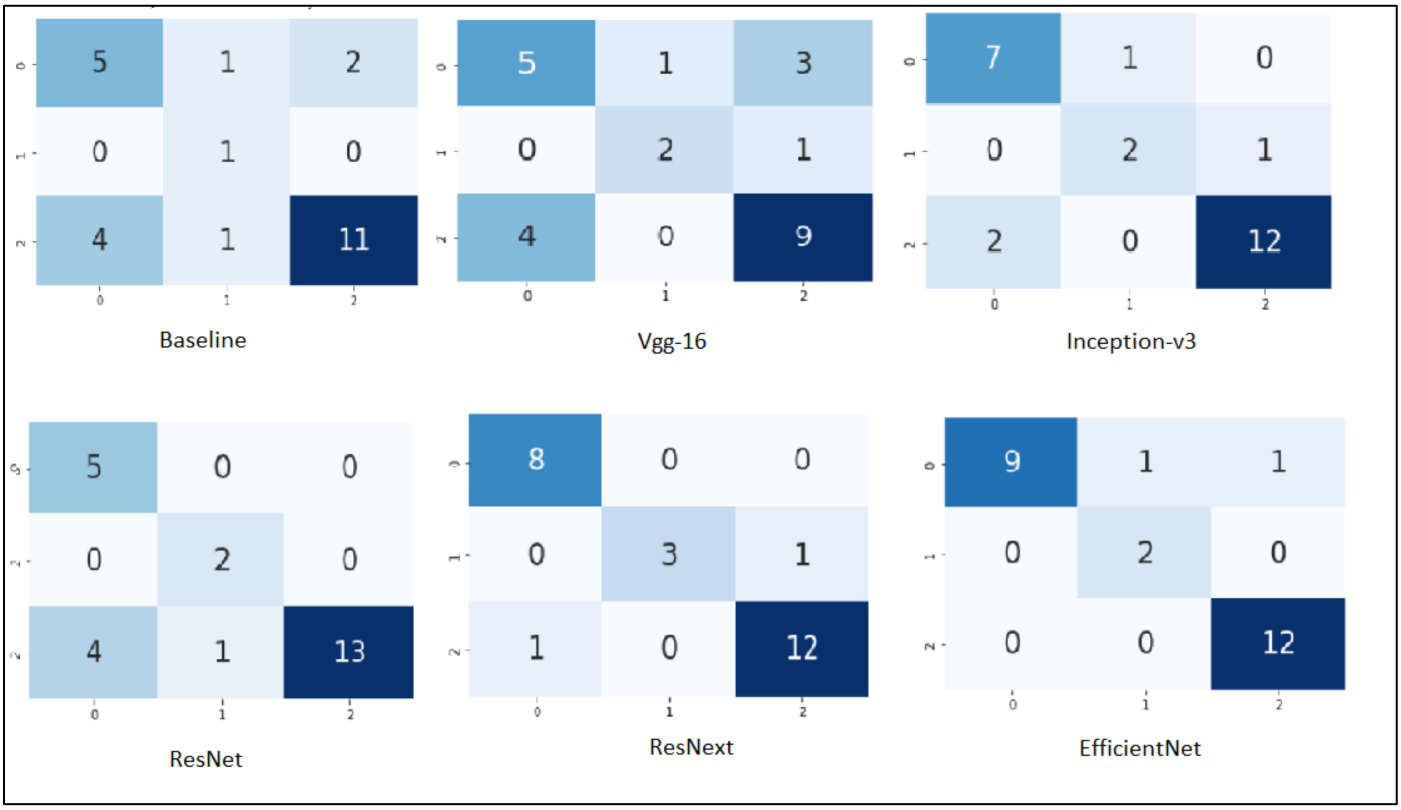

Fig 16 Confusion Matrix (Models Trained on Augmented Data)

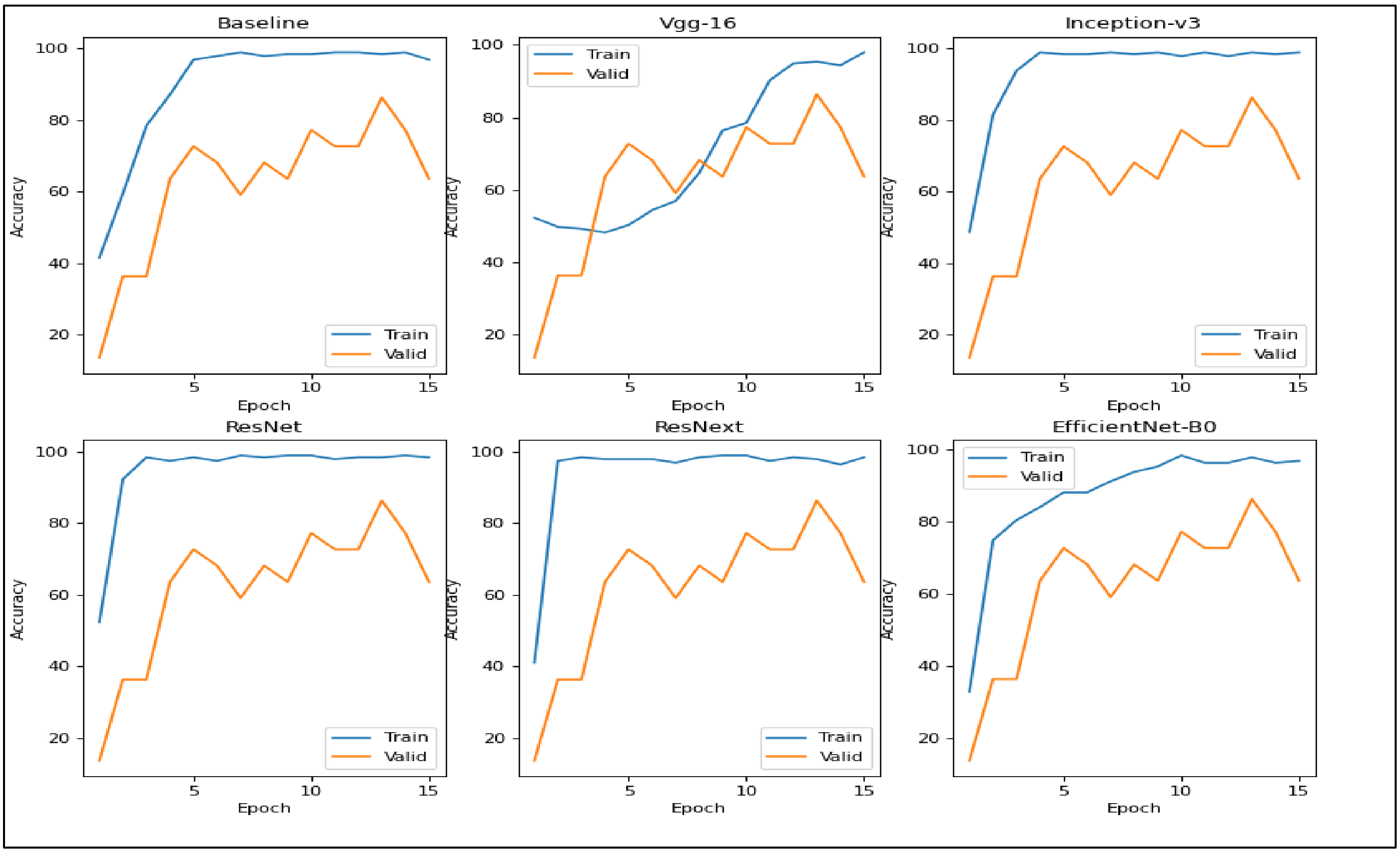

Fig 17 Evolution of the Accuracy during Training and validation Steps.
(Models Trained on Original Data)

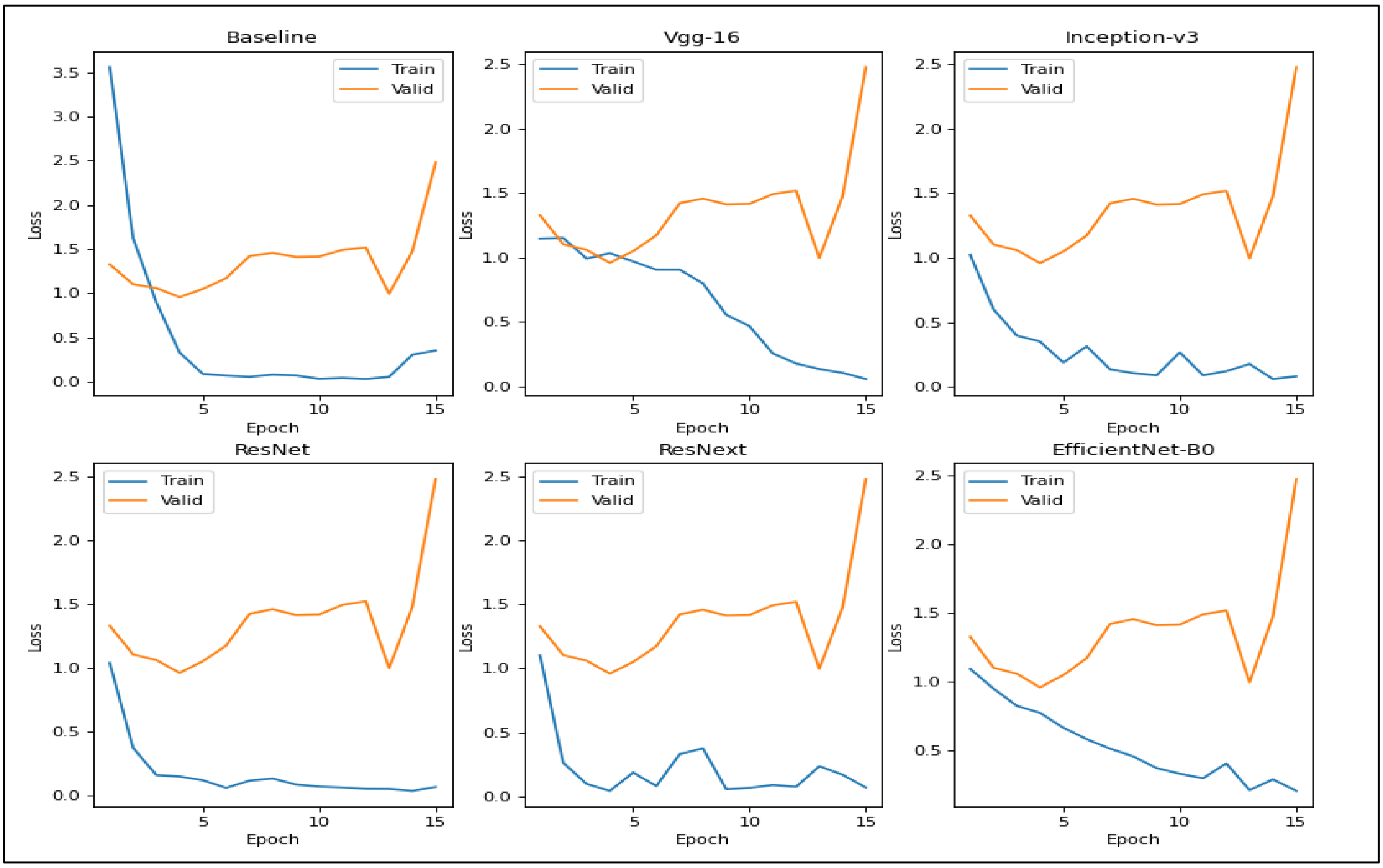

Fig 18 Evolution of the Loss during Training and Validation Steps.
(Models Trained on Augmented Data).

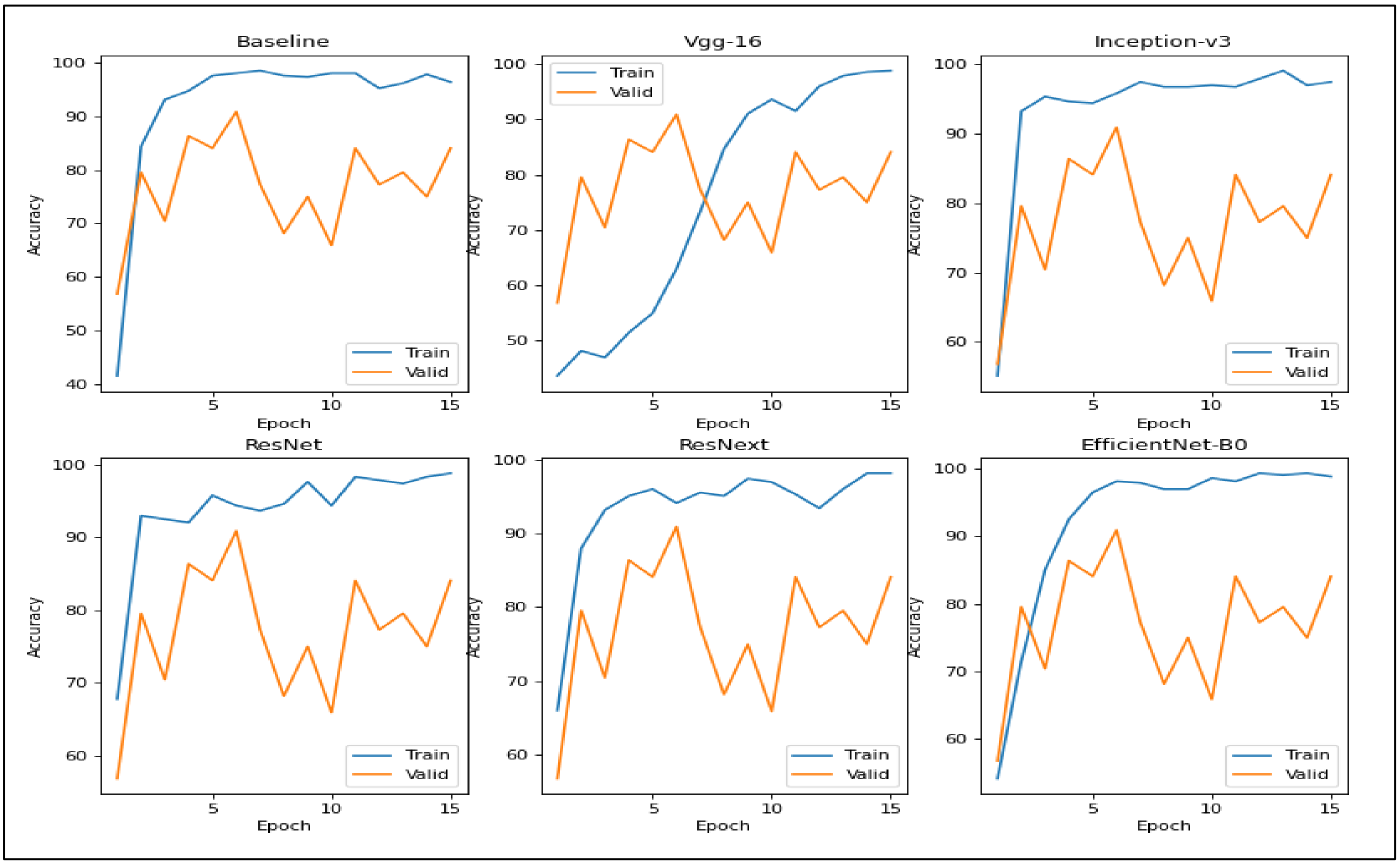

Fig 19 Evolution of the Accuracy during Training and Validation Steps.
(Models Trained on Augmented Data)

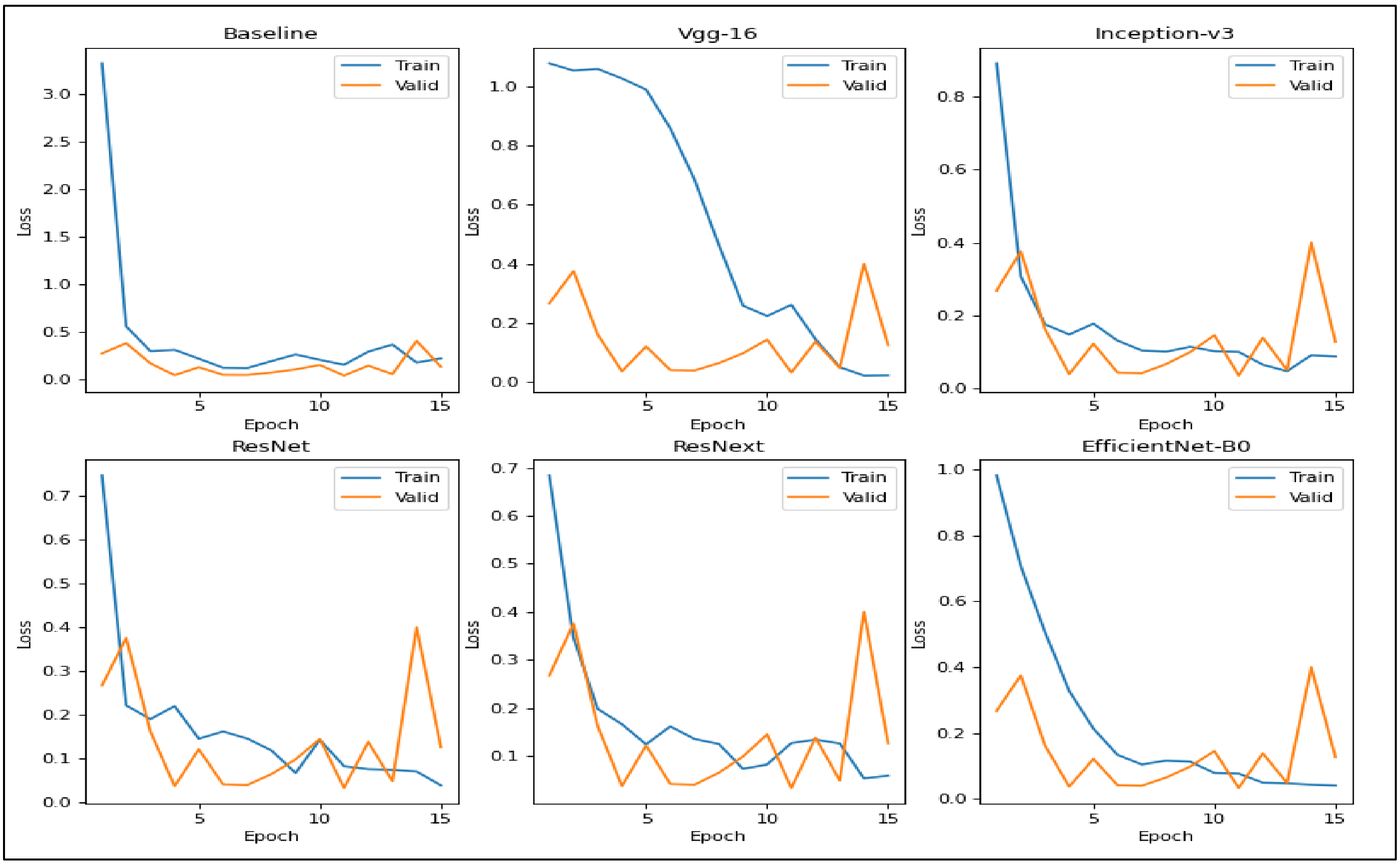

Fig 20 Evolution of the Loss during Training and Validation steps.
(Models Trained on Augmented Data).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

- *Evaluation of Models Trained on Original Data*

As can be seen in the Table 1, EfficientNet model outperforms all models in term of accuracy and F1-score when the data is not augmented, despite it has a less complex architecture with respect to other models (ResNet, ResNext and inception_v3). Figure 18 and 19 show the stability during the training of the EfficientNet model (validation black curves), while we observe fluctuations during the training of the other models.

Baseline and Vgg16 models show lower performance. This is excepted since it has the smaller number of CNN blocks so they extract less discriminant features.

The dataset is imbalanced, where the second class (2 years sheep) is the minority class. Confusion matrices depicted in Figure 15 show that all models find difficulties to recognize this class, where 1 out of 3 are correctly classified.

- *Evaluation of Models Trained on Augmented Data*

All models in this experiment are trained on augmented data as describing in the previous section. Table 2 shows that the ResNext and EfficientNet outperform the other models while ResNext outperforms slowly the EfficientNet in the term of F1-score. The good news is the data augmentation make models recognize the minority class. Confusion matrices illustrated in Figure 16 and F1-score column in Table 2 prove this improvement.

The huge difference between the train and validation values in curves on Figures 17 and 18 has become minimal in Figures 19 and 20; the curves became more convergent; and this means that the augmentation technique succeeded in reducing the overfitting.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER SIX
DEPLOYING THE DEEP LEARNING MODEL

In order to deploy our best model so that they can be used on a global scale, a mobile application should be the best solution. As a first stage a web application is created to deploy our project using the Flask which is a web development framework developed in Python [18]. In this section, we are going to describe the steps followed to complete this task.

➢ *App Design*

To run the app, the following steps are followed

- Save the best model using the pickle library.
- Setting up the environment to deploy the model with Flask [18] and creating the working tree directory as in Figure 21.

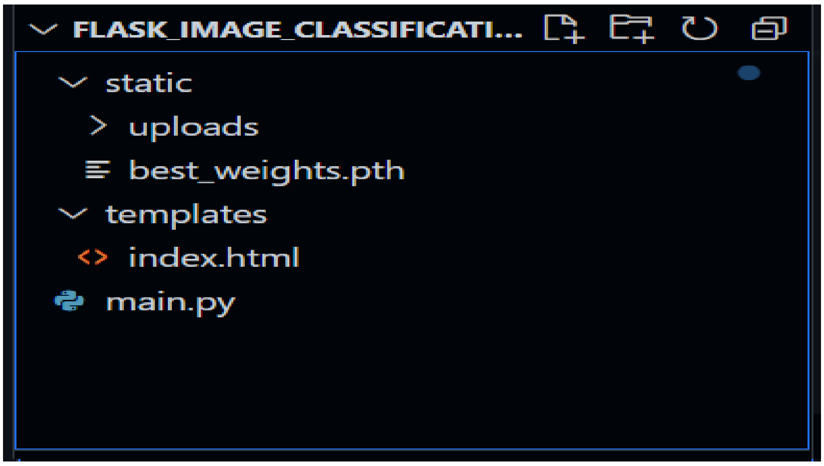

Fig 21 Flask App Working Tree Directory.

- *Static* folder: weights of the best model and all images stored in the sub folder *Uploads.*
- *Templates* folder: holds any template to be used in the app. In our case, one template is created, the home page.
- *main.py*: contains the flask routines and python functions needed to run the app. Figure 22 presents the most of these as explained in the following :

```
> class MyModel(torch.nn.Module): ···

  model_path = "static/best_weights.pth"
  model = MyModel(3)
  model.load_state_dict(torch.load(model_path))
  # Since we are using our model only for inference, switch to `eval` mode:
  model.eval()

> def preprocess(IMG_SIZE=224): ···
  transform = preprocess()

> def get_category(model, image_class_mapping, image_path): ···

> def get_prediction(model, image_class_mapping, image_file): ···

  @app.route('/')
  def home():
      return render_template('index.html')

  @app.route('/', methods=['POST'])
> def upload_image(): ···

  @app.route('/display/<filename>')
> def display_image(filename): ···

  if __name__ == "__main__":
      app.run()
```

Fig 22 Python Functions and Flask Routines in the main.py.

- *MyModel* class: a pytorch class to define the architecture of our model. The model is initialized by instantiating this class. Then the best weights are loaded.
- *request.files()* found in *def upload_image()* collects teeth sheep image given user inputs.
- Image collected is pre-processed by *def preprocess()*
- *def predict()* is responsible for predicting the age of the input sheep image and return one category :{ “1 year”, “2 years”, “3 years”}
- The *def predict() also called the def display_image()* function to display the image and the prediction value.
- Flask routines is responsible to collect variable like image and the predicted category and display them in the home page.
- Finally, we run the app and see the website’s development and environment status in our local device. The followings figures show the running app.

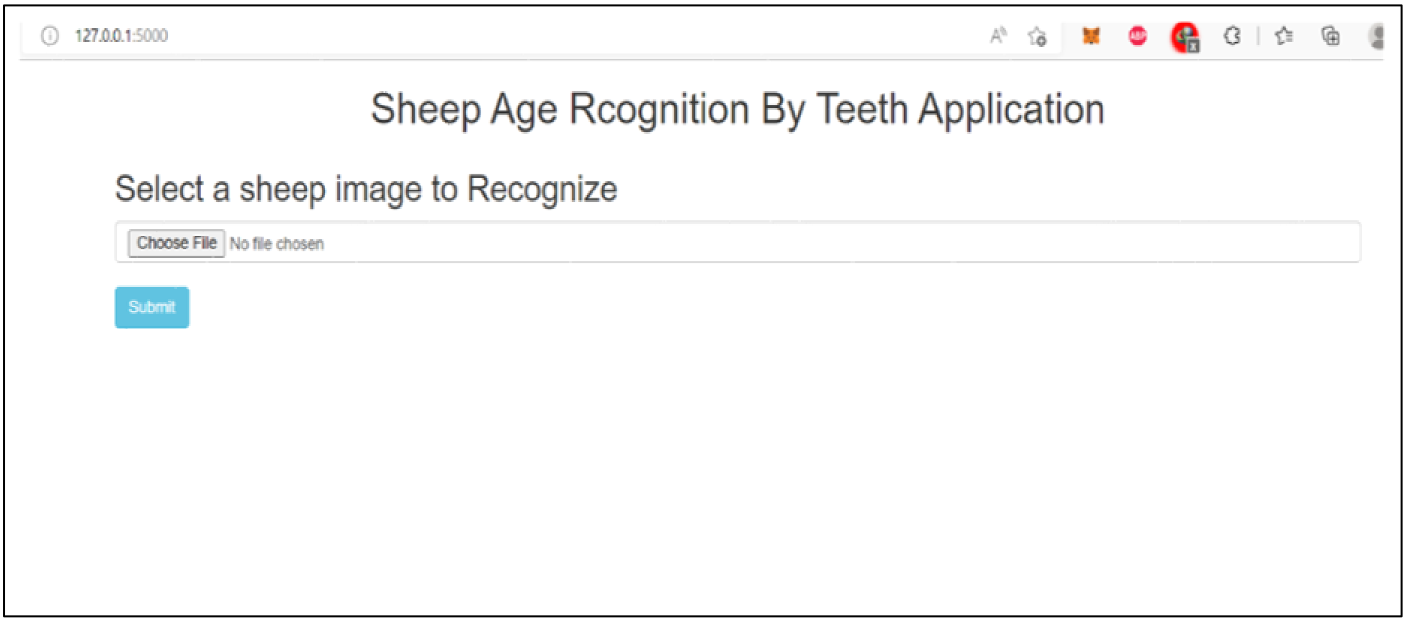

Fig 23 Home Page before Model Deployment.

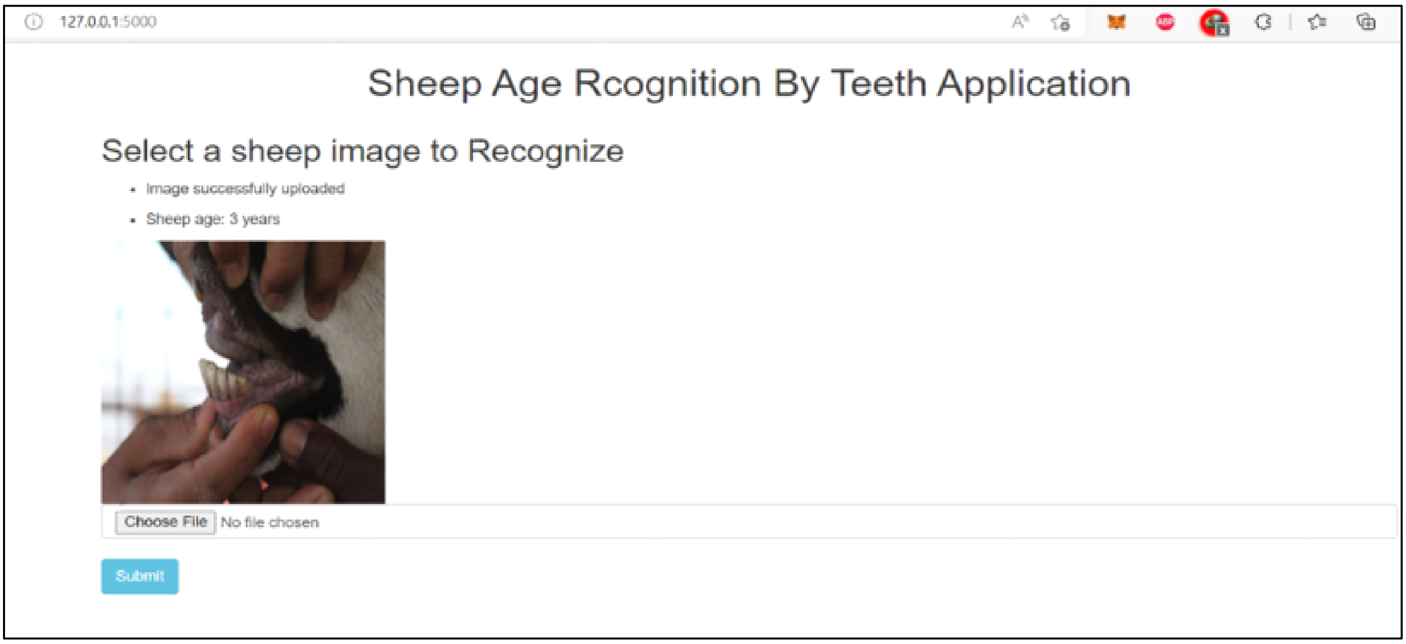

Fig 24 Home Page after Model Deployment.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

CHAPTER SEVEN
CONCLUSION AND FUTURE WORK

In this work, we conducted a two set of experiments in which applied six deep learning models baseline, Vgg16, Inception_v3, ResNet, ResNext and EfficientNet. Except the baseline model, we applied the pretrained versions of these models on augmented and not augmented data. Results obtained depicted that EfficientNet outperforms the other models. The augmented data brings the suitable improvement, where the models became able to recognize the minority class.

As a future work, I will collect more data to develop the project. I will also study the possibility of applying few-shot learning for our task which is a method that doesn't require many data. A mobile app which we can use anywhere will be more practical than a web app.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24SEP371

REFERENCES

[1]. Geoff Casburn, Sheep Development Officer, Wagga Wagga. July 2016, Primefact 1481, second edition.

[2]. R. Azad, B. Azad, and I. T. Kazerooni, "Real-time and robust method forhand gesture recognition system based on cross-correlation coefficient", Adv. Comput. Sci., Int. J., vol. 2, no. 5/6, pp. 121–125, Nov. 2013.

[3]. X. Chen, L. Zhang, T. Liu, and M. M. Kamruzzaman, "Research on deep learning in the field of mechanical equipment fault diagnosis image quality", Journal of Visual Communication and Image Representation, Vol. 62, pp. 402–409, Jul. 2019.

[4]. A. Abyaneh, A. Foumani and V. Pourahmadi, "Deep Neural Networks Meet CSI-Based Authentication", arXiv Organization, Cornell university, Nov. 2018.

[5]. M. S. Hossain and G. Muhammad, "Emotion recognition using secure edge and cloud computing", Information Sciences, Vol. 504, No. 2019, pp. 589–601, Dec. 2019.

[6]. D. Melinte and L. Vladareanu, "Facial Expressions Recognition for Human–Robot Interaction Using Deep Convolutional Neural Networks with Rectified Adam Optimizer", MDPI Journal, Vol. 20, Issue. 13, Apr. 2020.

[7]. Q. Zhu, P. Zhang, Z. Wang and X.Ye, "A New Loss Function for CNN Classifier Based on Predefined Evenly-Distributed Class Centroids", IEEE Transaction and Journals, Vol. 8, pp. 10888-10895, Dec. 2019.

[8]. A. Abdelhady, AE. Hassanenin, A. Fahmy A. "Sheep identity recognition, age and weight estimation datasets". arXiv.org perpetual, pp. 1-7, 2018.

[9]. H. Raadsma, I. Harris, D. Tao, M. Khatkar, J. Gao, S. Thompson, W. Gibson and M. Ferguson "Artificial Intelligence in Wool Production", pp. 1-44, 2019.

[10]. C. Ma, X. Sun, C. Yao, M. Tian, L. Li , "Research on Sheep Recognition Algorithm Based on Deep Learning in Animal Husbandry", Journal of Physics: Conference Series.

[11]. K. Simonyan, and A. Zisserman, " Very Deep Convolutional Networks for Large-Scale Image Recognition". The 3rd International Conference on Learning Representations (ICLR2015), 2015.

[12]. https://github.com/tensorflow/models/blob/master/research/slim/nets/inception_v1.py.

[13]. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in CVPR, 2016.

[14]. S. Xie, R.B. Girshick, P. Dollar, Z. Tu and K. He, " Aggregated Residual Transformations for Deep Neural Networks ", journal CoRR, volume abs/1611.05431, 2016.

[15]. Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International conference on machine learning (pp. 6105-6114). PMLR.

[16]. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Advances in Neural Information Processing Systems 32 [Internet]. Curran Associates, Inc. p. 8024–35, 2019.

[17]. M. Chao, S. Xiaolin, Y. Chunxia, T. Minglu, L. Linyi, 'Research on Sheep Recognition Algorithm Based on Deep Learning in Animal Husbandry', Journal of Physics: Conference Series, 2020.

[18]. M . Grinberg, "Flask web development: developing web applications with python.", " O'Reilly Media, Inc.", 2018.