Unique Dog Identification Using Convolutional Neural Network

Supriya Telsang, Rohan Sasne, Riya Loya, Rohit Mandake, Tejas Rokade, Rounak Lohe

Department of Engineering, Sciences and Humanities (DESH)
Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract —
A CNN model is developed using deep learning that proved successful in predicting the distinctive face of a pet dog. In order to train a model on a relatively small dataset without losing the ability to generalise, features found from high-resolution pictures of vast datasets are applied. This method is called transfer learning. The State-of-the-Art VGG16 model, which gave it the necessary foundation, enabled this classification to be done. To create our model, which accurately distinguished between our one-of-a-kind object and other objects, two dense neural network layers were added. It also incorporated the Augmentation feature in the training phase that randomly used to shift the orientation of random training data in order to address the issue of randomization of location in the stance of any object, which allowed to enhance the precision of our project. This model was successful in achieving a 96.88% accuracy rate.

Keywords *— Convolutional Neural Network, Deep Learning, Facial Recognition, Transfer Learning, VGG16.*

I. INTRODUCTION

The Area of Image classification has been widely researched upon and many Deep Learning models have been developed in the past which proved significantly accurate in terms of predicting the image classes and distinguishing between the objects. Image categorization is still a crucial computer approach even if it is the oldest. There has been a considerable shift from Fourier transforms to neural networks. The obstacles in the photos, such as position fluctuations, occlusion, illumination, concealment, and more, mean that it is still a difficult computation. [1]. In our model, transfer learning is utilized to properly categorize a distinctive face of a unique pet dog by precisely extracting the distinctive elements from the photos. The ability to transfer knowledge between different tasks is innate in humans. We use the knowledge we gain while conducting one particular task to solve other similar activities in the same way. A learner can apply knowledge or abilities they have mastered in one environment to another situation through the cognitive process known as "Transfer." [2]. Traditionally, algorithms for machine learning and deep learning were developed to function for a certain feature-space distribution. [2]Models must be completely rewritten when the feature-space distribution changes, and collecting the necessary training data takes time. Since deep learning models require sufficient labelled data for training. As a result, building a machine learning-based model for a target domain with little labelled data for supervised learning is particularly difficult. Transfer learning would significantly boost the learning's effectiveness in these circumstances. Transfer learning's major goal is to improve the performance of a machine learning system in the target domain by applying labelled data or information that has been gleaned from related domains.

The study of transfer learning is motivated by people's capacity to successfully apply previously acquired information to resolve new problems more rapidly or efficiently. [2]. In an NIPS-95 workshop on "Learning to Learn," the fundamental justification for transfer learning in the field of machine learning was presented. [2]. Schroff et al. from Google, Inc. proposes a system called FaceNet. FaceNet learns to represent a face in Euclidean space so that the separation between faces can be determined by its distance. [2]. When extracting features, they make use of Deep Convolutional Network. [2]. The experiment's accuracy results from the LFW and YouTube Faces dataset are 99.63% and 95.12%, respectively. [2]. Additionally, Q Wen et al. trained and employed pre-trained CNN network architecture, such as Inception-V3, VGG-Face, VGG-19, and VGG-16 model architecture, and they presented deep learning model architecture for FER on three (3) public datasets (CK+, JAFFE, and FACES). [3]. These strategies and tactics were effective in identifying facial expressions using the transfer learning methodology. [3].

The backbone of our model is the State of the Art ‘VGG16’ Deep learning model, where Convolutional Neural Networks are used to extract the features out of images. Convolutional Neural Networks (CNN), in particular, are a proven image representation and classification method for image analysis and applications. [4]. When employing CNN, deep learning does not require a difficult way to extract the features. The outcomes from datasets trained by CNN will vary depending on the architecture and the datasets. [4]. Modern outcomes can be specifically attained by convolutional neural networks that have been trained on vast amounts of labelled data. [5].

In this research paper, we continue to identify unique dogs using a pre-trained CNN network architecture. The fundamental distinction between our work and others is the use of convolutional neural network (DCNN) architecture and the direct training of the network on the given dataset. We apply the transfer learning technique to further increase the effectiveness of our model and prevent over-fitting as well. By employing this method with the vgg-16 network design, network performance will be improved and computer complexity will be reduced. From the vgg-16 model, feature vectors were retrieved, and the convolutional block weights were all frozen. A new dense layer with two layers was added after the final vgg-16 convolutional layer, and the output was given to a new classifier for categorization. The trial results show that our recommended technique can result in test accuracy up to 96.88%.

II. METHODOLOGY/EXPERIMENTAL

Initially, convolutional neural networks (CNNs) are used to categorise images. Layers such as convolutional, pooling, flattening, and fully connected layers are applied to the input images. [2]. After creating CNN models from scratch, fine-tuning of the model using the technique of image augmentation is done. One of the pretrained models, the VGG-16, is used to categorise images and assess accuracy for both training and validation data.

A) Theory:

➢ Convolutional Neural Network

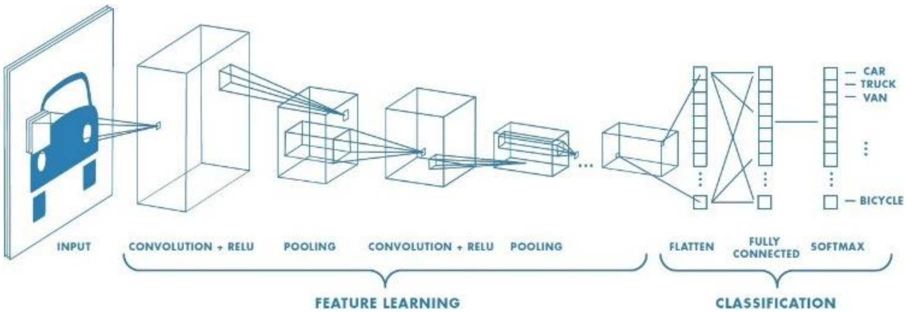

Fig. 1. Building blocks of a convolutional neural network

Convolutional neural networks are a type of artificial neural network that analyze picture inputs using multiple perceptrons and contain learnable bases and weights for different properties of images that can be used to differentiate them from one another. Because some parameters are shared, convolutional neural networks benefit from the local spatial coherence in the input images and can employ fewer weights as a result. In terms of complexity and memory, this strategy works well. A convolutional neural network's fundamental components are as follows:

- Convolution Layer: The convolutional layer processes the input matrix over the kernel matrix to produce a feature map for the following layer. The convolution mathematical method involves sliding the kernel matrix across the input matrix. Every site multiplies the matrix element by element and then adds the results to the feature map.
  In several disciplines, including image processing, statistics, and physics, convolution is a specific kind of linear operation that is commonly used. Convolution may be used on more than one axis.

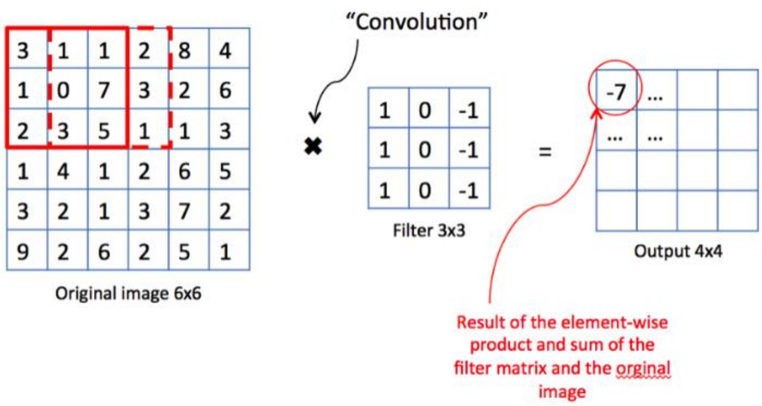

Fig. 2. Results from element-by-element matrix multiplication and summarization are plotted on a convolutional layer feature map.

- Non-linear activation functions (ReLU): Is one kind of non-linear adjustment we make to the input signal. The activation function comes after the convolutional layer as a node. A piecewise linear function called the rectified linear unit activation function (ReLU) will output the input if it is positive and 0 otherwise.

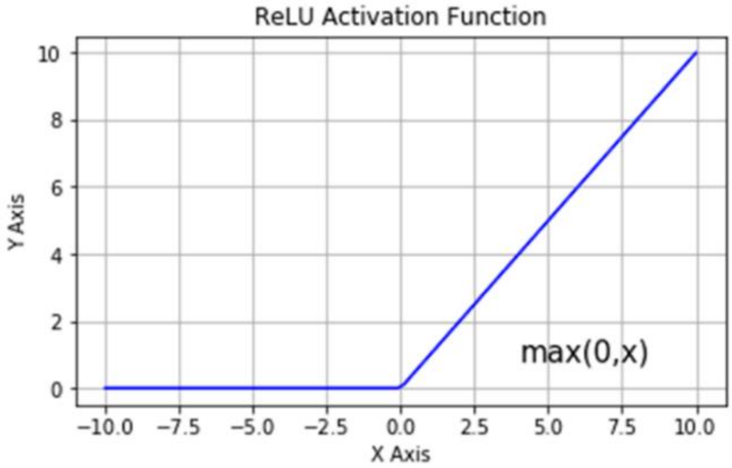

Fig. 3. ReLU activation function

- Pooling Layer - The fact that the feature map of the convolutional layer captures the precise positions of input features is a drawback. This implies that even little adjustments to the source image, such as cropping or rotation, will result in a completely new feature map. We use convolutional layer down sampling to try to fix this issue. After the nonlinearity layer, a pooling layer can be added to perform down sampling. The representation becomes roughly invariant to modest input translations with the aid of pooling. The majority of the pooled outputs are said to be translation-invariant if a little amount of the input is translated.

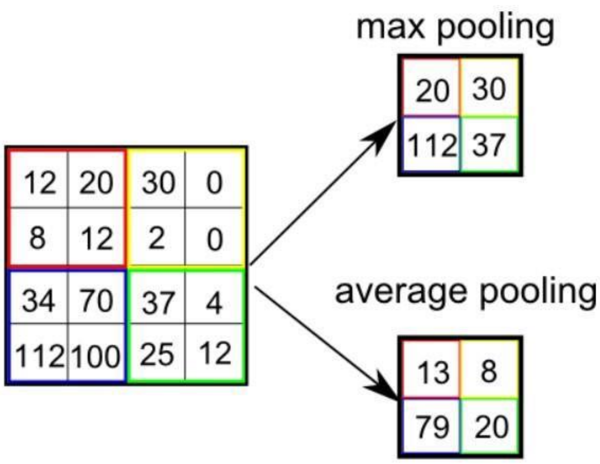

Fig. 4 Pooling types include maximum pooling and average pooling.

- Fully Connected Layer: In a convolutional neural network, the output of the last pooling layer is used as the input for the fully connected layer. These levels could be one or several. Before the system may be said to be fully connected, each second-tier node must be connected to the first layer.

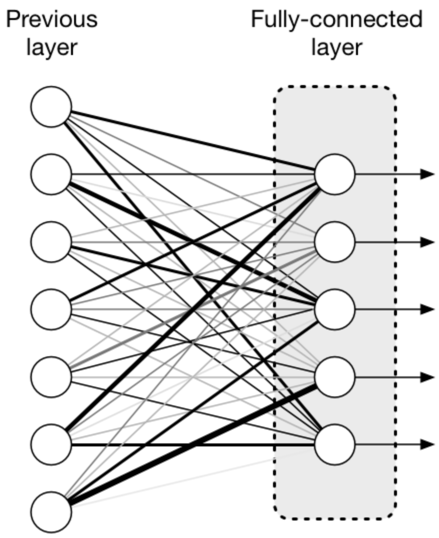

Fig. 5 Fully connected layer

➢ VGG-16 Model:

The Architecture of VGG-16 is as follows:

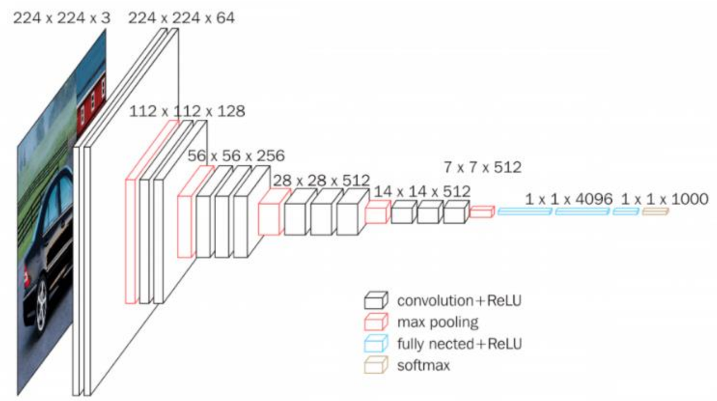

Fig. 6 VGG-16 models' architecture, it has 1 SoftMax classifier and 2 fully connected layers among its 13 convolutional layers.

VGG-16 Model - In their 2014 paper, "Very Deep Convolutional Network for Large Scale Image Recognition," Andrew Zisserman and Karen Simonyan made the initial presentation of the VGG-16 architecture.
A 16-layer network with convolutional and fully linked layers was developed by Karen and Andrew. For simplicity, only 33 convolutional layers were used, placed on top of one another The architectural network of the VGG-16 model is described in detail in Fig. 6 as follows:

❖ The first and second convolutional layers are composed of 64 feature kernel filters with a 33-filter size. As the input image (a RGB image with a depth of three) is communicated via the first and second convolutional layers, the dimensions change to 224x224x64. The output is then transferred with a stride of two to the maximum pooling layer. [2].

- ❖ The third and fourth convolutional layers' 124 feature kernel filters have a 33-filter size. [2]. These two layers are followed by a max pooling layer with stride 2, and the output is subsequently reduced to 56x56x128. [2].
- ❖ The fifth, sixth, and seventh layers are convolutional layers with kernel sizes of 33 and 256 feature maps, respectively. A max pooling layer with stride 2 follows these layers. [2].
- ❖ Eighth through thirteenth are split into two sets of 512 kernel-based convolutional layers with a 33 kernel each. A max pooling layer with a stride of one follows these layers. [2].
- ❖ The sixteenth layer is a softmax output layer with a size of 1000 units. The fourteenth and fifteenth levels are fully connected hidden layers with a size of 4096 units. [2].

➢ Pre-trained models and the use of Transfer Learning:

A research project called ImageNet aims to create a sizable library of photographs with annotations, such as labels for the pictures. ImageNet, which offers a range of picture classifications, has already been used to train a number of pre-trained models, including VGG-16, VGG-19, etc. These models were created from scratch and were trained using powerful GPUs on millions of photos that belonged to a huge number of different image categories. The model has acquired a strong knowledge of low-level properties like forms, edges, rotations, and illumination as a result of its massive training data. These characteristics can be used as feature extractors for new images in a range of computer vision challenges and can be shared to promote knowledge transfer.

```
Model: "vgg16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 block1_conv1 (Conv2D)       (None, 224, 224, 64)      1792

 block1_conv2 (Conv2D)       (None, 224, 224, 64)      36928

 block1_pool (MaxPooling2D)  (None, 112, 112, 64)      0

 block2_conv1 (Conv2D)       (None, 112, 112, 128)     73856

 block2_conv2 (Conv2D)       (None, 112, 112, 128)     147584

 block2_pool (MaxPooling2D)  (None, 56, 56, 128)       0

 block3_conv1 (Conv2D)       (None, 56, 56, 256)       295168

 block3_conv2 (Conv2D)       (None, 56, 56, 256)       590080

 block3_conv3 (Conv2D)       (None, 56, 56, 256)       590080

 block3_pool (MaxPooling2D)  (None, 28, 28, 256)       0

 block4_conv1 (Conv2D)       (None, 28, 28, 512)       1180160

 block4_conv2 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

 block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

 block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

 block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________
```

Fig. 7 VGG-16 Model Details

➢ Dataset:

This study is aimed to develop a robust unique object image classification model with constraints on the number of training examples for the unique dog and other animal species (binary classification problem). To apply constraints to the input photographs, we downscaled the number of images. For training, we collected 161 data photographs of Our Special Dog and other animals, and we collected 240 images for validation. In order to classify the images as belonging to either our one-of-a-kind pet dog or another category of animal, we use the pretrained VGG-16 model as a feature extractor. We learned new level characteristics tailored for the target problem while transferring low-level elements like edges, corners, and rotation.

Fig. 8 Train images of Unique Dog and other animals.

B) Design:

A crucial step is added before adding our new layers to the pre-trained model. The pre-trained layers of the model are frozen. This indicated that the base layers from the previously learned model will not be updated during training. Instead, for the new classification, updated the new layers by adding at the end. This is done to keep the learning from training on the Image Net dataset. At this point, if they weren't frozen, loss of important data would take place. Later, through a procedure known as fine-tuning, defrosted and trained these layers.

Setting trainable on the model to "False" will instantly freeze the foundation layers.

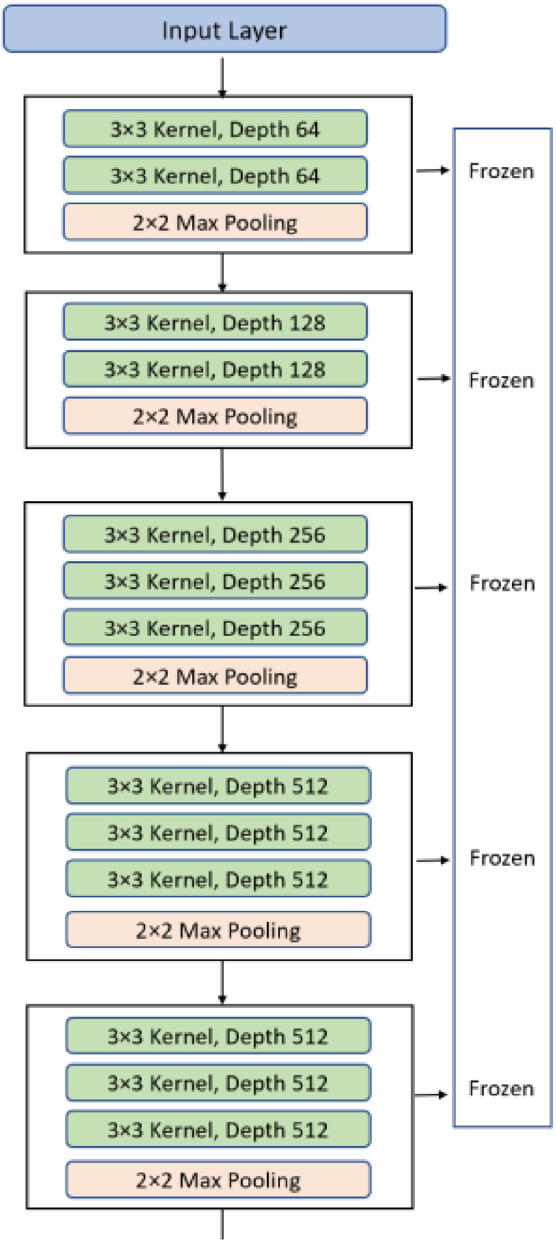

Fig. 9 The architecture of the VGG-16 is depicted in a block diagram. The weights of all 5 convolutional blocks are frozen in order to extract feature vectors from the VGG-16 model.

The pre-trained model was then given the new trainable layers. The features from the previously trained layers will be transferred to these new layers, which will use the new dataset to generate predictions. Two more layers have been added to the model. The pooling layer, which is the initial layer, reduces the size of the feature maps, the number of parameters that must be learned, and the amount of network processing necessary. The last layer that was added determines whether the image is of a Unique Dog or not.

C) Pseudo Code:

```
inputs = keras.Input(shape=(224, 224, 3))

# Separately from setting trainable on the model, we set training to False

x = base_model(inputs, training=False)

x = keras.layers.GlobalAveragePooling2D()(x)

# A Dense classifier with a single unit (binary classification)

outputs = keras.layers.Dense(1)(x)

model = keras.Model(inputs, outputs)
```

Fig. 10 Code for Addition of Two Dense layers to the CNN Network.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 224, 224, 3)]     0

 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 global_average_pooling2d (G  (None, 512)              0
 lobalAveragePooling2D)

 dense (Dense)               (None, 1)                 513

=================================================================
Total params: 14,715,201
Trainable params: 513
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig. 11 The CNN network after the addition of 2 new layers.

After including the new layers, assembled the model with options for loss and metrics. Here, we had to make some alternative decisions. Since the classification problem in our research was merely binary (Our Unique dog or not), binary cross entropy was chosen. Additionally, substitution of binary accuracy for conventional accuracy is done. Since the dataset is small, it was crucial that we add more information. Minor adjustments to the already-existing images are made so that the model could see a larger range of pictures to learn from. Instead of only recalling the images it trains on, this let our model learn to recognize fresh pictures of Our Unique dog.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Create a data generator
datagen_train = ImageDataGenerator(
    samplewise_center=True,  # set each sample mean to 0
    rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
    zoom_range=0.1,  # Randomly zoom image
    width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
    height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
    horizontal_flip=True,  # randomly flip images
    vertical_flip=False,
)  # we don't expect our Unique dog to be upside-down so we will not flip vertically

# No need to augment validation data
datagen_valid = ImageDataGenerator(samplewise_center=True)
```

Fig. 12 Augmentation of the Input Images allowing our model to see a wider variety of images to learn from.

Any input image will be passed to the model in the following feature map:

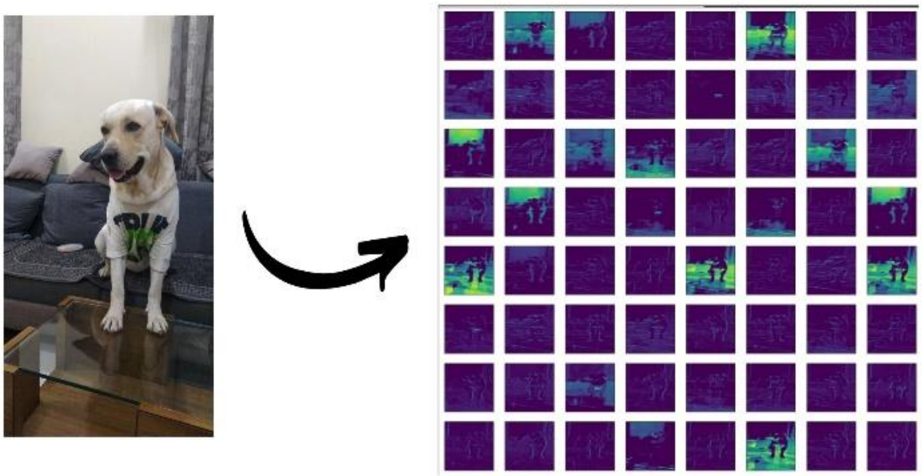

Fig. 13 Input image getting converted into features mapping.

The accuracy during training and validation both came out to be fairly high. Despite the fact that we could only train the model on a limited dataset, it was still able to achieve high accuracy and generalize effectively thanks to the information provided by the Image Net model. This indicated that it had a firm understanding of both other animals and our special pet dog.

When the model's additional layers were trained, we had a method to enhance it called fine-tuning. We completely unfroze the model and retrained it at a very low learning rate. The model was marginally improved as a result of the basic pre-trained layers' extremely modest movements and slight adjustments.

III. RESULTS AND DISCUSSIONS

The model metrics are as follows:

Using a pretrained model and image augmentation, CNN was fine-tuned, and model accuracy and loss were measured as follows:

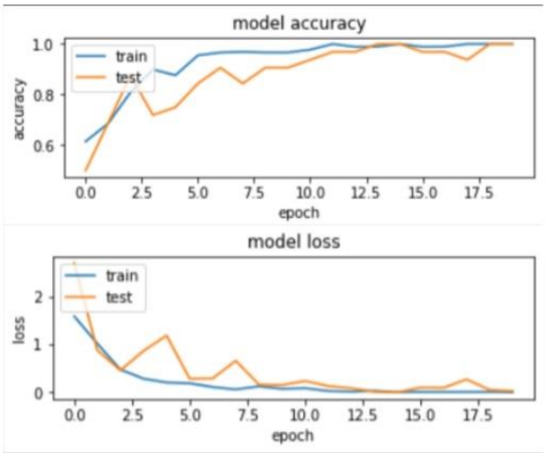

Fig. 14 Model accuracy and loss during CNN fine-tuning using a pre-trained model and picture augmentation.

To give an understanding of the project, our model was successfully able to classify the following two unique dogs of the same breed:

Fid. 15. Unique Recognition of two different dogs of the same breed.

The Result of our research is mentioned in the following table:

Cases	Training Accuracy	Validation Accuracy
Model performance with image augmentation: accuracy and loss	83.23%	81.40%
Model accuracy and loss during CNN fine-tuning using a pre-trained model and picture augmentation.	93.45%	96.88%

Two neural network models' training and validation accuracy are displayed in Table 16 in Figure 16.

With the help of photo augmentation, the basic convolutional neural network model's accuracy was increased to 81.40%. We use one of the pretrained models (VGG-16), which was trained on a big dataset of images and enhanced via image augmentation, to achieve accuracy of 96.88%.

IV. CONCLUSION

A cutting-edge machine learning software library from Google's Brain is called Tensor flow. Despite the large amount of training images, it is ideally suited for automatic classification of images. We concentrated on classifying 2 classes in this paper. In conclusion, using Tensor flow and the VGG-16 model as the project's core, we were able to accurately categorize the two types of animals with a 96.88% accuracy rate. By using Image Augmentation approach and fine-tuning the model, we have continued to work on lowering the misclassification rate. Our model's accuracy has increased as a result of this strategy. With the aid of embedded systems, we intend to use this image identification technique in smart home automation to use it as a security check for pets at home. We also plan to conduct additional research on this model's performance under various lighting conditions to further strengthen the training dataset. The proposed model will also be enhanced to support thermal imaging.

V. ACKNOWLEDGMENT

We want to thank Prof. Supriya Telsang, our adviser, for her great advice and encouragement as we worked on our research.

V. REFERENCES

[1] D. Meena, "An Efficient Framework for Animal Breeds Classification Using Semi-Supervised Learning and Multi-Part Convolutional Neural Network (MP-CNN)," *IEEE Access,* vol. 7, pp. 151783-151802, 2019.

[2] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *International Journal of Scientific and Research Publications (IJSRP),* vol. 9, no. 3, p. p9420, 2019.

[3] A. Chukwuemeka, "Transfer Learning Technique with VGG-16 for Near-Infrared Facial Expression Recognition," *Journal of Physics: Conference Series,* 2021.

[4] A. Bintang, "Face Recognition Using Light-Convolutional Neural Networks Based On Modified Vgg16 Model," *2019 International Conference of Computer Science and Information Technology (ICoSNIKOM),* pp. pp. 1-4, 2019.

[5] V. Pandey, "Image Classification Using Deep Neural Network," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN),* pp. 730-733, 2020.

[6] G. Chen, "Deep convolutional neural network based species recognition for wild animal monitoring," *2014 IEEE International Conference on Image Processing (ICIP),* pp. pp. 858-862, 2014.

[7] H. Yousif, "Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification," *2017 IEEE International Symposium on Circuits and Systems (ISCAS),* pp. pp. 1-4, 2017.

[8] Nguyen, "Animal Recognition and Identification with Deep Convolutional Neural Networks for Automated Wildlife Monitoring," *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA),* pp. 40-49, 2017.

[9] C. Wang, "Dog Breed Classification Based on Deep Learning," *2020 13th International Symposium on Computational Intelligence and Design (ISCID),* pp. 209-212, 2020.

[10] M. Pawar, "Transfer Learning for Image Classification," *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA),* pp. pp. 656-660, 2018.