COW BREED CLASSIFICATION

A
Report submitted in partial fulfillment of the requirement for
the
degree of

BACHELOR OF TECHNOLOGY
In
Computer Science & Engineering

By

ANKIT KUMAR (1901640100054)
SANA SIDDIQUI (1901640100233)
AVANISH DUBEY (1901640100083)
ATHARVA VISHWAKARMA (1901640100082)

Under the Supervision of

Mr. Atif Mahmood, Assistant Professor
Pranveer Singh Institute of Technology, Kanpur

Dr. A.P.J. Abdul Kalam Technical University Lucknow
2023

COW BREED CLASSIFICATION

A

Report submitted in partial fulfillment of the requirement for the degree of

B. Tech.

In

Computer Science & Engineering

Under the Supervision of

Mr. Atif Mahmood (Assistant Professor)

By

Ankit Kumar (1901640100054)
Sana Siddiqui (1901640100233)
Avanish Dubey (1901640100083)
Atharva Vishwakarma (1901640100082)

Pranveer Singh Institute of Technology, Kanpur
Dr. A.P.J. A.K. Technical University
Lucknow
2023

DECLARATION

We hereby declare that this submission is our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Signature:
Name : Ankit Kumar
Roll No. : 1901640100054

Signature:
Name : Avanish Dubey
Roll No. : 1901640100083

Signature:
Name : Sana Siddiqui
Roll No. : 1901640100233

Signature:
Name : Atharva Vishwakarma
Roll No. : 1901640100082

Date :

Certificate

This is to certify that Project Report entitled "COW BREED CLASSIFICATION" which is submitted by **Atharva Vishwakarma**, **Avanish Dubey**, **Ankit Kumar**, and **Sana Siddiqui** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of **Computer Science and Engineering** of **Pranveer Singh Institute of Technology**, affiliated to **Dr. A.P.J. Abdul Kalam Technical University, Lucknow** is a record of the candidates own work carried out by them under my/our supervision. The project embodies the result of original work and studies carried out by the students themselves and the contents of the project do not form the basis for the award of any other degree to the candidate or to anybody else.

Signature:

Dr. Vishal Nagar
Dean
CSE Department,
PSIT, Kanpur

Signature:

Mr. Atif Mahmood
Assistant Professor
CSE Department,
PSIT, Kanpur

Date :

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B.Tech. Project undertaken during B.Tech. Final Year. We owe special debt of gratitude to our project supervisor Mr. Atif Mahmood, Department of Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur for his constant support and guidance throughout the course of our work. His sincerely, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor Dr. Vishal Nagar, Dean, Department of Computer Science & Engineering, Pranveer Singh Institute of Technology, Kanpur for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:
Name : Ankit Kumar
Roll No. : 1901640100054

Signature:
Name : Avanish Dubey
Roll No. : 1901640100083

Signature:
Name : Sana Siddiqui
Roll No. : 1901640100233

Signature:
Name : Atharva Vishwakarma
Roll No. : 1901640100082

ABSTRACT

Cow breed classification is an important task for dairy farmers, animal researchers, and cow enthusiasts. However, manually identifying the breed of a cow can be challenging, time-consuming, and prone to errors.

In this project, we propose a machine learning approach to automatically classify cows into different breeds based on their images. We use a convolutional neural network (CNN) model that can learn the distinctive features of each breed from a large dataset of cow images. We also develop a website that allows users to upload their own cow images and get the predicted breed as well as some information about it. Our website also provides a gallery of cow images from different breeds.

To achieve this, we start by collecting a diverse dataset of cow images, encompassing various breeds, angles, and lighting conditions. We preprocess the dataset by resizing and normalizing the images to ensure consistency. Next, we train a CNN model using popular deep learning frameworks such as TensorFlow. The model architecture consists of multiple convolutional layers, pooling layers, and fully connected layers.

The trained model is then integrated into a Django-based web application to facilitate user-friendly access. The website enables users to upload cow images and receive real-time predictions regarding the breed of the cow in question. The backend of the web application handles the image processing and passes the input through the CNN model for classification. The predicted breed is displayed to the user with the characteristics and general information of the cow breed.

Throughout the development process, we leverage the powerful tools and libraries provided by Django, such as Django REST Framework and Django templates, to create an interactive and visually appealing user interface. The website also incorporates features like image previews, error handling, and data persistence to enhance the user experience.

The performance of the cow breed classification model is evaluated using various metrics, including accuracy, precision, recall, and F1 score. We conduct experiments with different hyperparameters and network architectures to optimize the model's performance. The final results demonstrate the effectiveness of the proposed CNN-based approach in accurately classifying cow breeds.

We evaluate our model on a test set of cow images and compare it with some baseline methods. We show that our model achieves high accuracy and robustness in cow breed classification.

Overall, this project showcases the integration of a CNN model for cow breed classification into a Django-based web application. It demonstrates the potential of machine learning techniques for solving practical problems in the agricultural domain, and provides a user-friendly interface for farmers, researchers, and cow enthusiasts to identify cow breeds accurately.

Our project contributes to the field of computer vision and machine learning by providing a novel and practical application of image classification. It also benefits the cow community by offering a convenient and fun way to learn about cow breeds.

TABLE OF CONTENTS

CHAPTER NO. TITLE **PAGE NO.**

LIST OF FIGURES

LIST OF ABBREVATION

CNN	Convolutional Neural Network
HTML	Hypertext Markup Language
API	Application Programming Interface
JSON	Java Script Object Notation
ML	Machine Learning
SQL	Structured Query Language
OS	Operating System
UI	User Interface
RAM	Random Access Memory
ROM	Read Only Memory

CHAPTER 1
INTRODUCTION

1.1 Motivation

Cows are one of the most common and important domestic animals in the world. They provide humans with various products such as milk, meat, leather, and fertilizer. They also have cultural and religious significance in some regions. According to the Food and Agriculture Organization (FAO), there are about 1.5 billion cows in the world, belonging to more than 800 breeds. Each breed has its own characteristics, such as size, color, shape, horns, milk production, and adaptability to different environments. Knowing the breed of a cow can help farmers to optimize their management practices, such as feeding, breeding, health care, and marketing. It can also help researchers to study the genetic diversity, evolution, and conservation of cows. Moreover, it can help cow enthusiasts to appreciate the beauty and diversity of cows.

To overcome these limitations, we propose a machine learning approach to automatically classify cows into different breeds based on their images. Machine learning is a branch of artificial intelligence that enables computers to learn from data and perform tasks that are difficult or impossible for humans. Image classification is one of the most common and popular applications of machine learning. It involves assigning a label to an image based on its content. For example, image classification can be used to recognize faces, objects, scenes, emotions, and diseases. In our project, we use image classification to recognize cow breeds.

1.1.1 Unknown History

Written records on the history of cattle older than the 18th century are scant or do not exist. We do know that most European breeds are not older than the period of the industrial revolution, when systematic selective breeding started. Many so-called 'land cattle breeds' or 'land races' are ascribed an ancient origin or advertised as: "known in the region since times immemorial" but are relatively new.

1.1.2 Gradual Differences Between Breeds

Differences between breeds are not as absolute as between species, as for instance the clear-cut differences between cattle, yak and bison. Breeds not only originated relatively recently from a common gene pool, but genetic isolation is rarely absolute.

Even the demarcation of zebu and taurine cattle, which evolved from two different sources and are clearly different in morphology, adaptation and behavior, is arbitrary since many intermediate types are known and several breeds have been developed by taurine-indicine crossbreeding.

1.1.3 Changes Over Time

Several breeds are different now from what they were only 20 years ago. In fact, selective breeding has accelerated the evolution of cattle to the point that the last two centuries saw more changes in appearance and production than the preceding millennia. Breeding objectives are not fixed, but follow changes, for example new preferences and requirements of consumers. By the late 19th century, Dutch-Friesian cattle were of a large, refined, single purpose dairy type; in the 1930's they were mainly of a stronger, coarser type; and in the 1950s they were of a small, deep bodied dual-purpose type. Today pure Dutch-Friesians are of a medium, milky dual-purpose type. In Holstein-Friesian, selection of the quantity of milk has changed to a preference for high protein content.

1.2 Background of the Problem

Identifying the breed of a cow is not an easy task. It requires expert knowledge and experience in cow anatomy and morphology. It also requires access to reliable sources of information, such as books, databases, or websites. Even with these resources, human judgment can be subjective, inconsistent, and influenced by factors such as lighting, angle, distance, and occlusion. Furthermore, some breeds are very similar to each other and can only be distinguished by subtle differences. Therefore, manual cow breed identification can be challenging, time-consuming, and prone to errors.

There are over 1000 breeds of cattle recognized worldwide, and they belong to two main species: Bos taurus and Bos indicus. Some breeds are selected for meat, milk, or draught purposes, while others are adapted to different climates and environments. Some breeds have distinctive features such as horns, humps, or coat patterns that can help distinguish them, but others may look very similar or have variations within the same breed.

We use a convolutional neural network (CNN) model for our image classification task. A CNN is a type of neural network that consists of multiple layers of neurons that can process images in a hierarchical manner.

Cattle breeds are groups of animals that share common characteristics and are genetically related. They have been classified based on various criteria such as cranial or horn morphology, coat color, geographic origin or molecular markers.

Classification of cattle breeds contributes to our understanding of the history of cattle and is essential for an effective conservation of genetic diversity.

Our project that can classify the cow breeds from images could have many applications and benefits, such as:

- Helping armers and veterinarians to monitor the health and productivity of their cattle.
- Providing information and education to consumers and the public about the diversity and origin of cattle breeds.
- Supporting conservation efforts for rare and endangered breeds.
- Enhancing research and innovation in animal genetics and breeding.

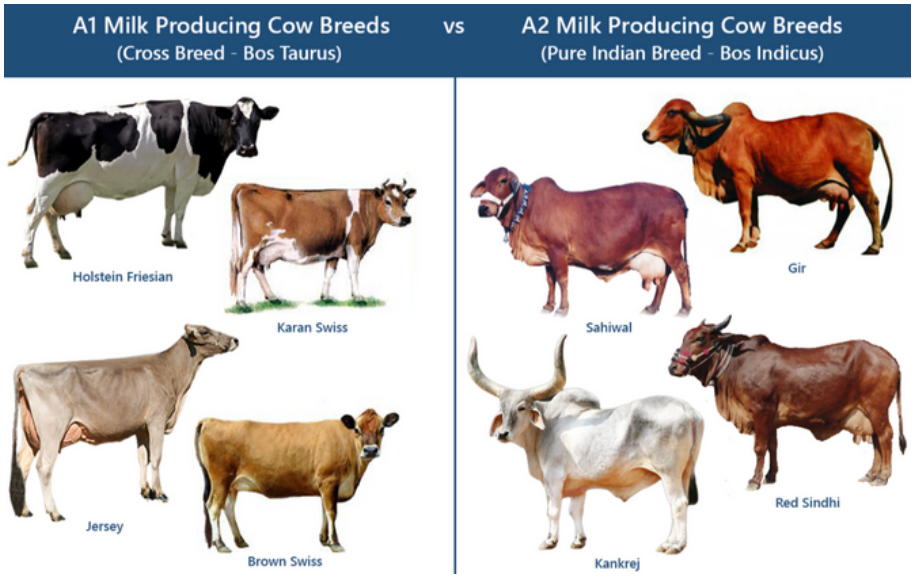

Fig 1.1 Cow Breed Clustering as per Application [7]

1.2.1 Current System

The current system for this problem is based on manual identification and classification of cow breeds by experts or farmers, using visual cues such as horns, humps, coat color and pattern, body size and shape, etc.

However, this system is prone to errors, inconsistencies, and subjectivity, especially when dealing with similar or mixed breeds. Moreover, it requires a lot of time and effort to collect and record the data for each cow.

Some studies have proposed computer vision-based approaches for automatic detection of dairy cow breeds, using image processing and machine learning techniques.

These approaches aim to extract relevant features from cow images, such as texture, shape, color, and key points, and use them to train and test different classifiers, such as support vector machines (SVM), k-nearest neighbors (KNN), artificial neural networks (ANN), etc. However, these approaches are still limited by the availability and quality of the data sets, the accuracy and robustness of the classifiers, and the generalization ability to new breeds or environments.

1.2.2 Issues in Current System

- It is not reliable or consistent, as different experts or farmers may have different criteria or opinions for identifying and classifying cow breeds.
- It is not efficient or scalable, as it requires a lot of time and effort to collect and record the data for each cow, and it may not be feasible for large or diverse herds.
- It is not accurate or comprehensive, as it may miss some subtle or hidden features that are important for distinguishing cow breeds, or it may confuse similar or mixed breeds.
- It is not adaptable or flexible, as it may not be able to cope with changes in the environment, such as lighting, background, pose, etc., or with new or unknown breeds.

1.3 Problem Statement

Classification of the hundreds of cattle breeds orders a large, seemingly chaotic variety in both appearance and performance into a consistent scheme. Placing breeds and varieties into well-defined groups reveals relations between types, subtypes, breeds and varieties. This information may be relevant for various reasons:

- Relationships between breeds allow a reconstruction of their history. Lack of documentation on the history of cattle breeding has created room for unfounded fiction, which, once printed, has often been amplified into a general belief. For instance, the long-horned Salers cattle are assumed to have descended directly from local aurochs that are depicted with similar horns in the nearby caves of Lascaux, but molecular evidence shows a close relationship with Alpine cattle.
- A classification may point out the uniqueness of a breed, which may be relevant to conservation. For some 20 years there has been an increasing interest in the preservation of local breeds, not only because genetic diversity may become irreversibly lost, but also because the breeds are perceived to belong to the cultural and historic heritage.

- Breed classification will also promote a better appreciation of the value of local breeds, often adapted to their environment and suitable for extensive management. This would prevent a counterproductive introduction of highly productive breeds in regions suitable only for extensive management, which has been practiced since the mid-20th century on an unforgivably wide scale. Rehabilitation and revaluation of locally adapted breeds will not only result in a sustainable conservation, but also improve agricultural production under local conditions.

1.4 Proposed Work

The work holds the objective to classify Cow breeds using Deep Learning Models and to get specific information about that breed like the quantity of milk they give, the cost of follow-up, its market price if sold, the amount of meat we can get, and all the general information related to that breed. In our project, we focus on four types of species in which we would be training our deep learning model. Those breeds are:

- Gir
- Sahiwal
- Rathi
- Red Sindhi

This model will be using 3 different CNN models in which we can check or classify an image by choosing a respective model and the model would categorize the breed type. It also will give general information regarding that breed like:

- Cow's milk production.
- Average height and weight.
- Price of purchasing.
- Fat content in the milk.
- Origin of the breed

Fig 1.2 Gir [8]

Fig 1.3 Rathi [8]

Fig 1.4 Red Sindhi [8]

Fig 1.5 Sahiwal [8]

1.5 Organization of Report

1.5.1 Introduction (Chapter 1)

The introduction gives an idea about our project, which is Cow Breed Classification. It tells about the motivation behind undertaking this project, the background of the existing problem, and the problem statement. Also in this chapter, we have mentioned the proposed work and organization of the report.

This project proposes a machine learning project on cow breed classification with a website to upload cow images and get the cow breed name with its general static information. The main objectives of this project are to collect and preprocess a large and diverse data set of cow images, to extract and select relevant features from the images, to train and test different classifiers, to deploy the best classifier on a website, and to evaluate and improve the system.

1.5.2 Literature Review (Chapter 2)

The literature review provides an overview of the history and diversity of cattle breeds and their classification systems, as well as the existing computer vision-based approaches for automatic detection of dairy cow breeds.

It also discusses the challenges and opportunities of using machine learning techniques for cattle identification and detection. It shows that there is a need for an automatic and intelligent system that can identify and classify cow breeds from images with high accuracy and efficiency, which is the aim of this project.

1.5.3 Methodology (Chapter 3)

The methodology on which we have worked. It explains the flow of project and how in a sequential manner sticking to the plan we have worked on this project. And it's contain the implementation of the algorithm and the Inception v3 model that we have used to build the classifier. It also contains information on the implementation of the front end of the webapp.

The methodology for this project can be summarized as follows:

- Collect images of cows belonging to different breeds such as Gir, Sahiwal, Rathi and Red Sindhi from various sources such as online databases, websites, farms or cameras.
- Preprocess the images to enhance their quality and reduce noise, such as by resizing, cropping, rotating, filtering or augmenting them.
- Extract features from the images that can capture the phenotypic characteristics of the cows, such as their coat color, horn shape, ear size, forehead curvature, etc. This can be done using various techniques such as edge detection, color histogram, texture analysis or deep learning.
- Select the most informative features that can discriminate between different breeds using statistical methods such as fixation index (FST), principal component analysis (PCA) or random forest (RF).
- Train a classifier using the selected features and a labeled dataset of cow images with their corresponding breeds. The classifier can be based on various algorithms such as support vector machine (SVM), k-nearest neighbor (KNN), decision tree (DT) and artificial neural network (ANN).
- Evaluate the performance of the classifier using metrics such as accuracy, precision, recall or F1-score on a test set of cow images with unknown breeds.
- Integrate the ML model on the website that can allow users to upload their own images of cows. The website can also display some statistics or facts about the different breeds and their characteristics.

1.5.4 Results and Discussion (Chapter 4)

It contains information on the implementation of the test results i.e., the accuracy we received, the plots of accuracy, and loss functions for various numbers of epochs we ran to achieve maximum accuracy of our model.

Results: we present and analyze the results of the classifier design and evaluation step. We show the confusion matrix and the performance metrics of each classifier on the testing set. We also show some examples of correct and incorrect predictions made by each classifier. We use tables, charts, and images to illustrate the results.

Discussion: we discuss the findings and implications of the results. We compare the performance of different classifiers and identify the best one. We also discuss the strengths and weaknesses of each classifier and the possible reasons for their success or failure. We also discuss the limitations and challenges of the project and suggest some future directions for improvement.

1.5.5 Conclusion (Chapter 5)

We summarize the main points and contributions of the project. We restate the aim and objectives of the project and highlight the main findings and outcomes. We also emphasize the novelty and usefulness of the project and its potential applications and benefits for various stakeholders. We also acknowledge the limitations and challenges of the project and propose some future directions for improvement.

CHAPTER 2
LITERATURE REVIEW

Cow breed classification is an important task for various purposes, such as animal husbandry, dairy production, conservation, and research. However, the current system for this task is based on manual identification and classification by experts or farmers, which is unreliable, inefficient, inaccurate, and inflexible. Therefore, there is a need for an automatic and intelligent system that can identify and classify cow breeds from images using computer vision and machine learning techniques.

Several studies have reviewed the history and diversity of cattle breeds and their classification systems based on morphological, geographical, or molecular criteria. These studies have shown that there are over 1000 breeds of cattle recognized worldwide, and they belong to two main species: Bos taurus and Bos indicus.

These species are further divided into several groups or subgroups according to their origin, adaptation, or purpose. However, these classification systems are not consistent or comprehensive, and they may not reflect the true genetic relationships among breeds.

Some studies have proposed computer vision-based approaches for automatic detection of dairy cow breeds, using image processing and machine learning techniques. These approaches aim to extract relevant features from cow images, such as texture, shape, color, and key points, and use them to train and test different classifiers, such as support vector machines (SVM), k-nearest neighbors (KNN), artificial neural networks (ANN), etc.

However, these approaches are still limited by the availability and quality of the data sets, the accuracy and robustness of the classifiers, and the generalization ability to new breeds or environments. Some studies have also discussed the challenges and opportunities of using machine learning techniques for cattle identification and detection. These studies have highlighted the importance of data

collection and preprocessing, feature extraction and selection, classifier design and evaluation, and system deployment and improvement. They have also suggested some future directions for research and development in this field, such as using deep learning techniques, incorporating additional information sources, enhancing interpretability and feedback mechanisms, etc.

The conventional constructs of identifying animals can be categorized as:

1. Permanent Recognition Construct (PRC)
2. Semi-Permanent Recognition Construct (SRC)
3. Temporary Recognition Construct(TRC)

According to, the sketching pattern is applied for the recognition of cattle such as Holsteins and Guernsey with broken color. High drawing skills of an individual for sketching is needed which should be comparable to standard image quality and positively affect the cattle identification process. However, this method cannot be used for the identification of solid collared breeds such as Red Poll and Brown Swiss breed as some artificial marking methods such as ear tagging and tattooing that are discrimination based are needed. However, the method of ear tagging damages the cattle's ear in the long run.

As iterated in Petersen's work, the muzzle print-based cattle recognition method using blue ink and A-5 paper was the first attempt to get a permanent recognition method for cattle. In the method, skills are required to acquire the muzzle pattern's print image, by holding firm the cattle. Lately, the research community has shifted attention to advancing cattle recognition using the image of muzzle print as a new paradigm for cattle identification.

According to the print image the muzzle pattern is made up of beads and ridges patterns. Muzzle dermatoglyphics such as granola, ridges, and vibrissae from various breeds are not the same. Similarly, proposed in Mishra et al. is method of cattle breeds recognition using the beads and ridges features of muzzle print images. Similar to the work of Mishra et al. is Minagawa et al. they proposed a cattle identification method using muzzle print, the performance evaluation was made using filtering techniques for muzzle image analysis and morphological approaches.

Equal Error Rate (EER) of 0.419 was reported by them. Contrary to Minagawa, it is a framework proposed by Barry. The framework is a cattle recognition tool using muzzle print images. They reported the 241 false non match rates (FNMR) over 560 4 genuine acceptance rate (GAR) and 5197 false matches over 12,160 impostors matching closely with the same value of EER of 0.429, respectively.

In their cattle identification effort, Kim et al. proposed a method that could recognize the Japanese black cattle using the cattle face's pixel intensity. The results of the experiment were reported based on the image datasets of 4 cattle breeds used which were captured on A-5 paper with blue ink for the purpose of cattle recognition. Nevertheless, the performance of the matching refinement approach and the original SIFT approach were compared, and the value of EER equal to 0.0167 was achieved.

An object recognition method that is based on CNN was proposed in. The proposed architecture which combines RGB image and its corresponding depth image for object recognition is made up of two unconnected CNN processing streams, which are sequentially integrated with a late fusion network. ImageNet is employed for the training of the CNNs in which the depth image is encoded as a rendered RGB image, making the information that is contained in the depth data to go round over all the three RGB channels, and subsequently, a standard and pre-trained CNN is employed for the recognition.

Due to limited availability of large scale depth datasets that are labeled, CNNs that are pre-trained on ImageNet are employed. Proposed in, is another object recognition method, which employs deep CNN. Jingqui proposed the method of object recognition based on image entropy that was aimed at identifying the behavior of a cow object that is on the motion against a complicated background. Andrew et demonstrated the suitability of computer vision pipelines that utilize deep neural architectures to carry out automated Holstein Friesian cattle detection in addition to individual identification in a farm set up. In the process of extracting features from an image, Kumar, posited that pre-processing is important for object tracking accuracy but feature extraction and representation algorithms that are based on appearance are unable to perform the recognition of object as a result of image blurriness due to noise, low illumination and the unconstrained environment.

In summary, this literature review shows that cow breed classification is a complex and challenging problem that requires an interdisciplinary approach involving computer vision and machine learning techniques. It also shows that there is a gap in the literature regarding a comprehensive and robust system that can identify and classify cow breeds from images with high accuracy and efficiency. This project aims to fill this gap by proposing a novel machine learning project on cow breed classification with a website to upload cow images and get the cow breed name with its general static information.

CHAPTER 3
METHODOLOGY

3.1 CNN System Architecture

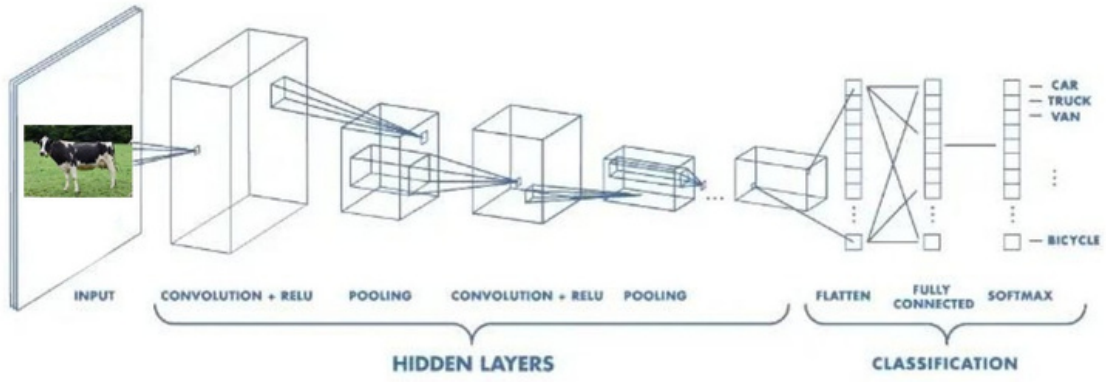

Fig 3.1 CNN System Architecture

3.1.1 Overview of CNN

A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be. The human brain processes a huge amount of information the second we see an image. Each neuron works in its own receptive field and is connected to other neurons in a way that they cover the entire visual field.

Just as each neuron responds to stimuli only in the restricted region of the visual field called the receptive field in the biological vision system, each neuron in a CNN processes data only in its receptive field as well. The layers are arranged in such a way so that they detect simpler patterns first (lines, curves, etc.) and more complex patterns (faces, objects, etc.) further along by using a CNN one can enable sight to computers.

3.2 Overview of Inception V3 Model

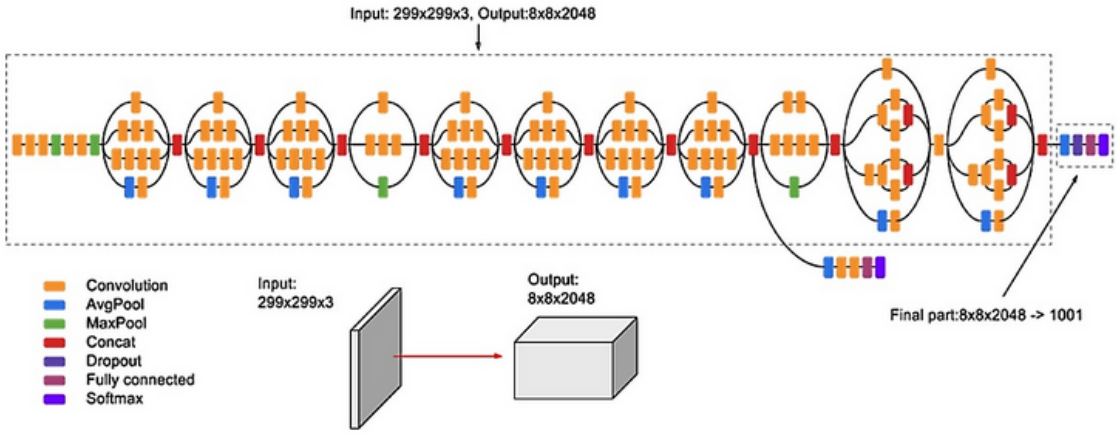

Fig 3.2 Inception V3 System Architecture

Inception v3 is an image recognition model that has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concatenations, dropouts, and fully connected layers. Batch normalization is used extensively throughout the model and applied to activation inputs. Loss is computed using SoftMax.

In comparison to VGGNet, Inception Networks (GoogLeNet/Inception v1) have proved to be more computationally efficient, both in terms of the number of parameters generated by the network and the economical cost incurred (memory and other resources). If any changes are to be made to an Inception Network, care needs to be taken to make sure that the computational advantages aren't lost.

Thus, the adaptation of an Inception network for different use cases turns out to be a problem due to the uncertainty of the new network's efficiency. In an Inception v3 model, several techniques for optimizing the network have been suggested to loosen the constraints for easier model adaptation. The techniques include factorized convolutions, regularization, dimension reduction, and parallelized computations.

3.3 Deep Learning

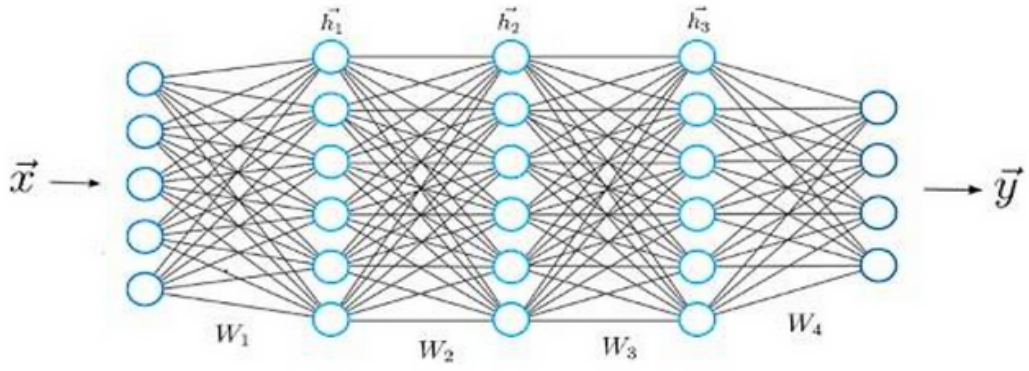

Fig 3.3 Deep Learning Architecture

Deep learning is a subfield of machine learning that involves the use of neural networks to model and solve complex problems. Neural networks are modeled after the structure and function of the human brain and consist of layers of interconnected nodes that process and transform data.

The key characteristic of deep learning is the use of deep neural networks, which have multiple layers of interconnected nodes. These networks can learn complex representations of data by discovering hierarchical patterns and features in the data. Deep learning algorithms can automatically learn and improve from data without the need for manual feature engineering.

Deep learning has achieved significant success in various fields, including image recognition, natural language processing, speech recognition, and recommendation systems. Some of the popular deep learning architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs).

3.4 Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or Plaid ML. Keras is widely used in image classification projects.

Keras is highly powerful and dynamic framework and comes up with the following advantages:

- Larger community support.
- Easy to test.
- Keras neural networks are written in Python which makes things simpler.
- Keras supports both convolution and recurrent networks.
- Deep learning models are discrete components, so that, you can combine into many ways.

For this cow breed classification project, we use Keras to build and train a convolution neural network (CNN) for classifying images of cows based on their breed.

3.5 TensorFlow

TensorFlow is an open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. TensorFlow was developed by the Google Brain team and is widely used in industry and academia.

TensorFlow provides a collection of workflows to develop and train models using Python, JavaScript, or Swift, and to easily deploy in the cloud, on-premises, in the browser, or on device no matter what language is use.

3.6 Transfer Learning

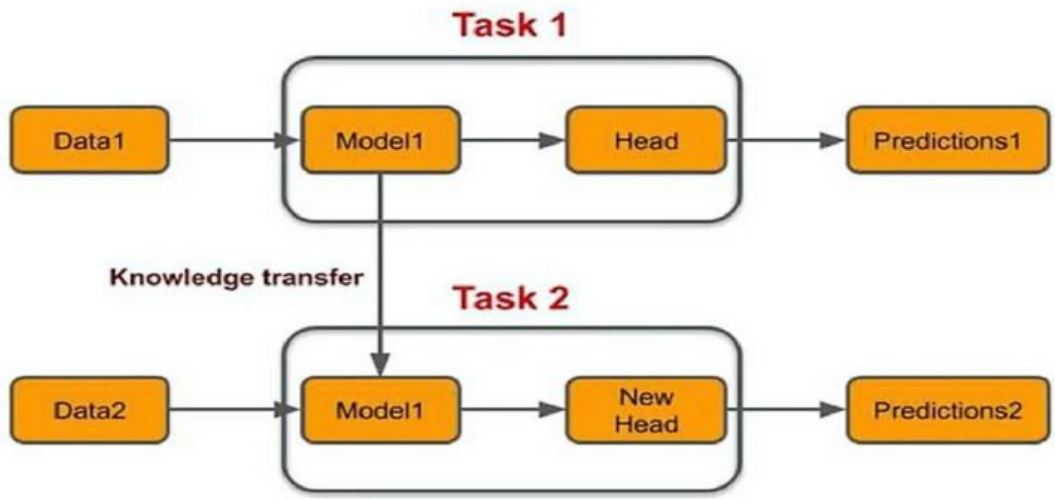

Fig 3.4 Transfer Learning Concept

Transfer learning is a machine learning technique that involves reusing a pre-trained model on a new problem. It is based on the idea that the knowledge gained from one task can be transferred to another related task, improving the performance and reducing the training time.

3.6.1 Why Transfer Learning?

Transfer learning is useful when the new task has limited data or is similar to the previous task. For example, if we want to train a model to recognize different types of animals, we can use a pre-trained model that was trained on a large dataset of images, such as ImageNet, and fine-tune it to our specific task. This way, we can leverage the features learned by the pre-trained model, such as edges, shapes, colors, etc., and adapt them to our new task.

Transfer learning is useful when the new task has limited data or is similar to the previous task. For example, if we want to train a model to recognize different types of animals, we can use a pre-trained model that was trained on a large dataset of images, such as ImageNet, and fine-tune it to our specific task. This way, we can leverage the features learned by the pre trained model, such as edges, shapes, colors, etc., and adapt them to our new task.

3.6.2 How Transfer Learning Works?

Transfer learning works by transferring the weights of a neural network that was trained on one task to a new neural network that will be trained on another task.

There are different ways to do this, depending on the similarity between the tasks and the amount of data available for the new task.

- **Training a Model to Reuse it :** One way to do transfer learning is to train a model on one task and then reuse it for another task. For example, we can train a model to detect faces and then use it for facial recognition. In this case, we can either use the whole model as it is or modify some parts of it according to our needs.
- **Using a Pre-Trained Model :** Another way to do transfer learning is to use an already pre-trained model that was trained on a large and diverse dataset, such as ImageNet. There are many pre-trained models available online, such as VGG, ResNet, Inception, etc., that can be used for various tasks. In this case, we can either use the pre-trained model as a feature extractor or fine-tune some or all of its layers.

3.7 Collecting Dataset & Performing Augmentation

3.7.1 Data Collection

In order to build intelligent applications capable of understanding, machine learning models need to digest large amounts of structured training data. Gathering sufficient training data is the first step in solving any AI-based machine learning problem. Data collection means pooling data by scraping, capturing, and loading it from multiple sources, including offline and online sources. High volumes of data collection or data creation can be the hardest part of a machine learning project, especially at scale. Furthermore, all datasets have flaws. This is why data preparation is so crucial in the machine learning process.

3.7.2 Data Augmentation

Data augmentation is a set of techniques to artificially increase the amount of data by generating new data points from existing data. This includes making small changes to data or using deep learning models to generate new data points.

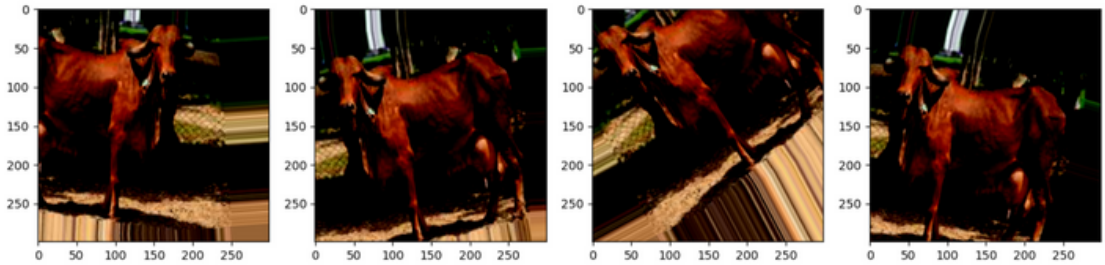

Fig 3.5 Images After Augmentation

Data augmentation can help improve the performance and accuracy of machine learning models by forming new and different examples to train datasets1. It can also reduce the reliance on collecting and labeling data, which can be costly and time-consuming1. Data augmentation can be applied to different types of data, such as audio, text, image, and video.

For image data: geometric or color space transformations, kernel filters, random erasing, mixing images.

3.8 Performed Label Encoding on the Image (One hot encoder)

Most machine learning tutorials and tools require you to prepare data before it can be fit to a particular ML model. One hot encoder is a technique that is used to convert categorical data into numerical data by creating a binary column for each category and assigning a value of 1 or 0 to indicate the presence or absence of that category.

Most machine learning tutorials and tools require you to prepare data before it can be fit to a particular ML model. One hot encoder is a technique that is used to convert categorical data into numerical data by creating a binary column for each category and assigning a value of 1 or 0 to indicate the presence or absence of that category.

One hot encoder is useful for data that has no relationship or order among the categories, such as colors or names. It avoids creating a false sense of priority or hierarchy among the categories that may affect the model performance.

Cow Breed
Rathi
Gir
Rathi
Sahiwal
Gir
Gir

Rathi	Gir	Sahiwal
1	0	0
0	1	0
1	0	0
0	0	1
0	1	0
0	1	0

Fig 3.6 Label Encoding on Image

One hot encoder is useful for data that has no relationship or order among the categories, such as colors or names. It avoids creating a false sense of priority or hierarchy among the categories that may affect the model performance.

However, one hot encoder may also increase the dimensionality of the data and create sparse matrices that may be inefficient to store and process12. Therefore, one hot encoder should be used with caution and depending on the nature of the problem and the data.

3.9 Split Dataset into Training, Testing and Validation

We need to split a dataset into train and test sets to evaluate how well our machine learning model performs. Dataset split is essential for an unbiased evaluation of prediction performance and model validation.

The dataset is usually divided randomly into three subsets:

- **Training set** is used to train or fit the model, such as finding the optimal weights or coefficients for the machine learning algorithm.
- **Validation set** is used to tune the model hyperparameters, such as the learning rate, regularization, or number of hidden layers, and to select the best model among different candidates.
- **Testing set** is used to evaluate the final model performance on unseen data and to estimate how well the model will generalize to new data.

A common ration for dataset split is 80% for training, 10% for validation, and 10% for testing, but this may vary depending on the project and size of the dataset.

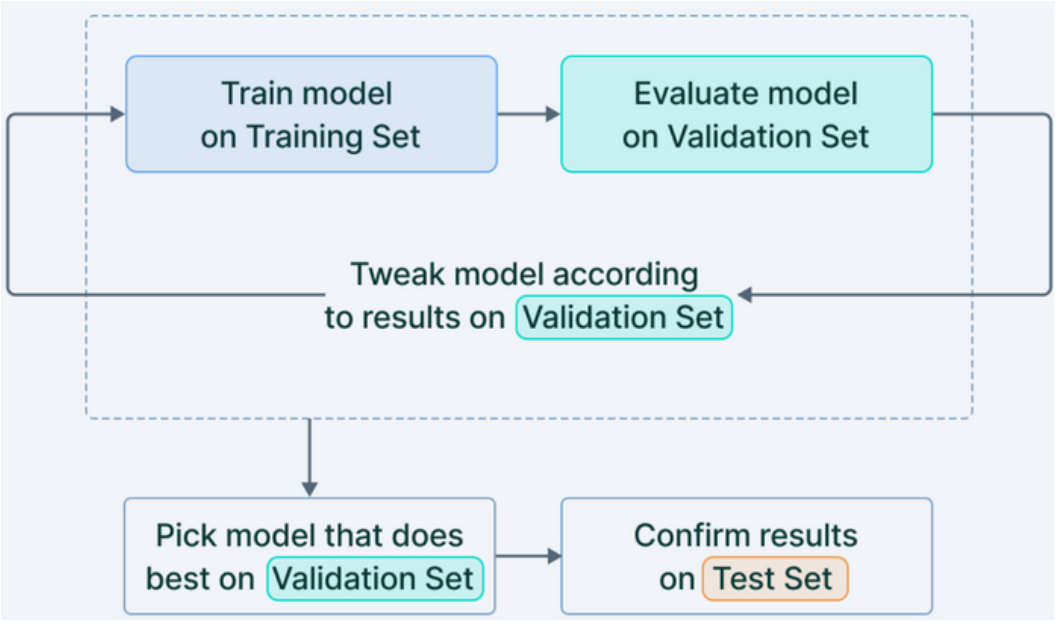

Fig 3.7 Training, Testing and Validation Dataset [11]

3.10 Building Model using Pre-trained Inception V3

3.10.1 Convolution Layer

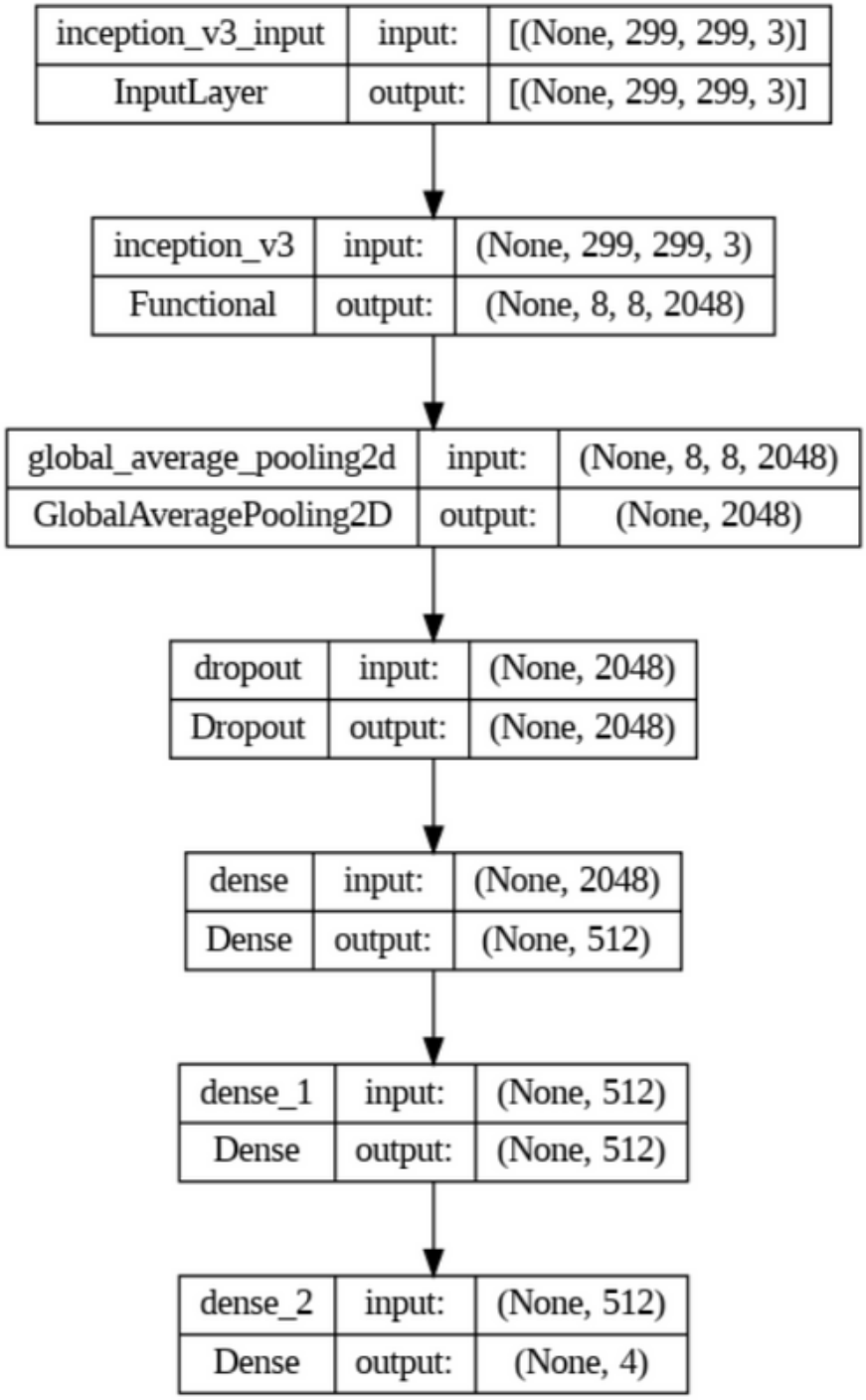

Fig 3.8.1 Layers added to Inception V3 Base Model

A convolution layer is a layer that applies a convolution operation to the input, which is a process of transforming an image by applying a kernel or a filter over each pixel and its local neighbors across the entire image. A convolution layer can extract features or patterns from the input image, such as edges, shapes, colors, or textures.

In this project, we have used a pre-trained Inception V3 base model to build our own model for cow breed classification. We have used the transfer learning technique, which means reusing the weights and features learned by the pre-trained model on a large dataset (such as ImageNet) and applying them to a new task with a smaller dataset (such as cow breed classification).

3.10.2 Pooling Layer (Max and Avg)

A pooling layer is a layer that performs a pooling operation, which is a form of down sampling that reduces the dimensions of the feature map. Pooling layers are used to make the model more robust to variations in the position of the features in the input image and to reduce the number of parameters and computations in the network.

There are two common types of pooling layers: max pooling and average pooling.

Max Pooling selects the maximum value in the region of the feature map covered by the filter. This way, it preserves the most prominent features in the region.

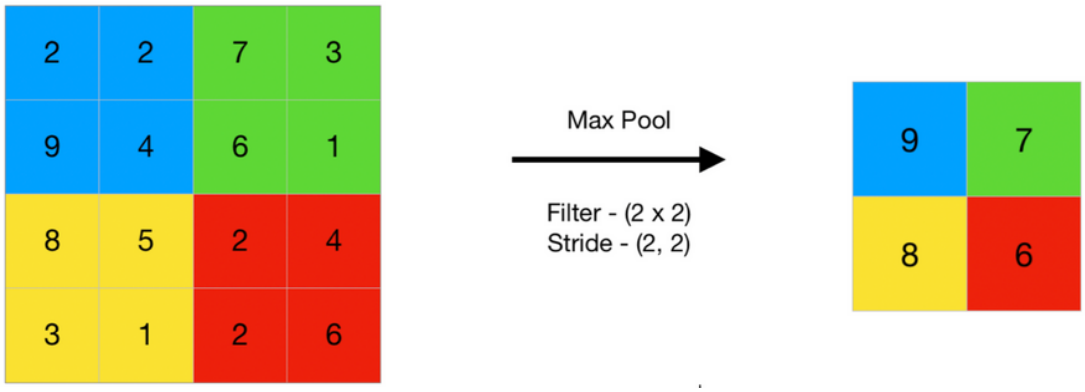

Fig 3.8.2.1 Max Pooling [12]

Average Pooling computes the average of the values in the region of the feature map covered by the filter. This way, it reflects the average presence of features in the region.

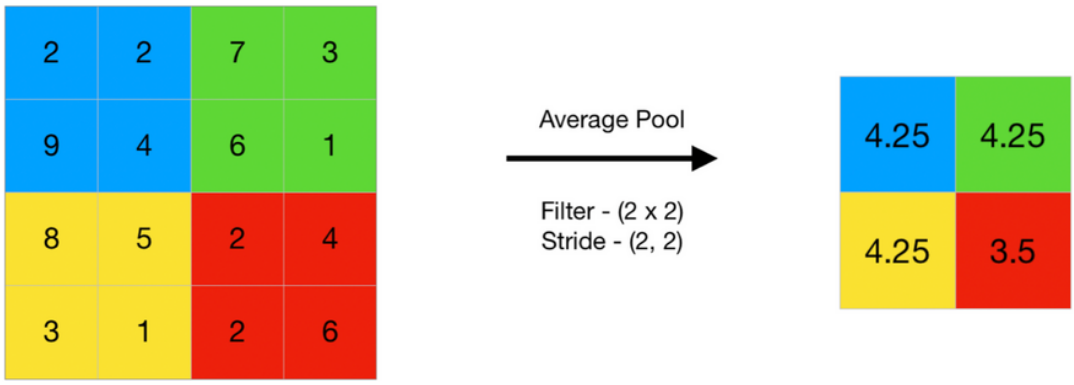

Fig 3.8.2.2 Average Pooling [12]

In this project, uses max pooling and average pooling layers after some convolution layers to reduce the size of the feature maps and extract the most relevant or average features from them. Using MaxPooling2D or AveragePooling2D layers in keras to implement max pooling or average pooling respectively.

3.10.3 Dropout Layers

Dropout layer is a layer that randomly sets some of the input units to zero with a given probability (called the dropout rate) during the training phase. This means that some of the neurons are temporarily ignored or dropped out from the network.

Dropout is a regularization technique that helps prevent overfitting in neural network models. Overfitting occurs when the model learns the noise or irrelevant patterns in the training data and performs poorly on new or unseen data. Dropout reduces overfitting by creating different versions of the network with different neurons dropped out, and averaging their predictions during inference. This makes the model more robust and less dependent on specific neurons or features.

In this project, we have used dropout layers after some dense or fully connected layers to reduce the complexity and size of the network and improve its generalization ability. To implement dropout with a given dropout rate using Dropout layers in keras.

Dropped out contribution to the activation of downstream neurons is temporarily removed on the forward pass, and any weight updates are not applied to the neuron on the backward pass.

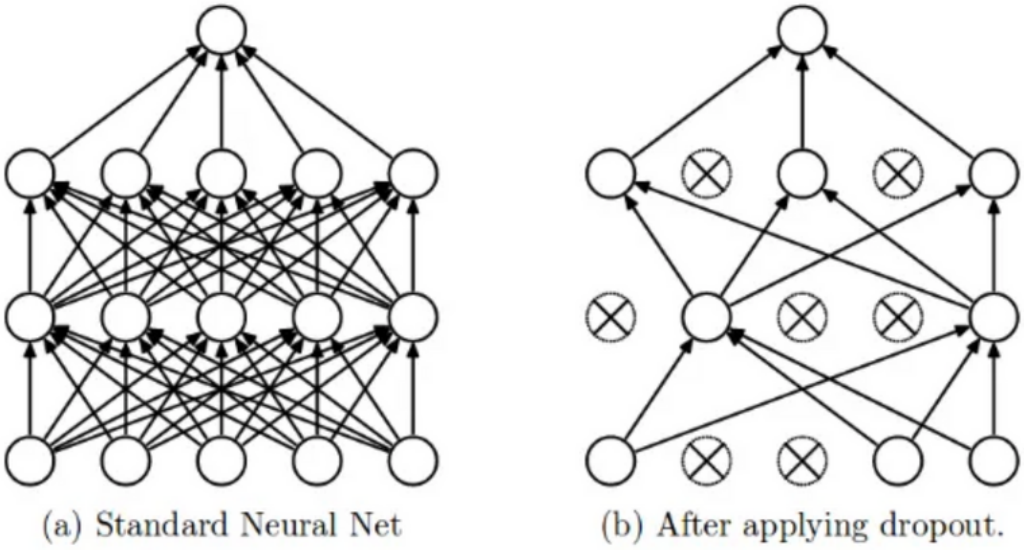

Fig 3.8.3 Dropping Layers (Drop out) [20]

3.10.4 Dense Layer

A dense layer is a layer of neurons that is fully connected to the previous layer, meaning each neuron in the dense layer receives input from every neuron in the previous layer. A dense layer performs a linear transformation on the input, followed by an optional activation function.

A dense layer can be represented by the following equation:

output = activation(dot(input, kernel) + bias)

where:

- **output** is the output vector of the dense layer.
- **activation** is the element-wise activation function (such as ReLU, sigmoid, etc.).
- **input** is the input vector from the previous layer.
- **kernel** is the weight matrix of the dense layer.
- **bias** is the bias vector of the dense layer.

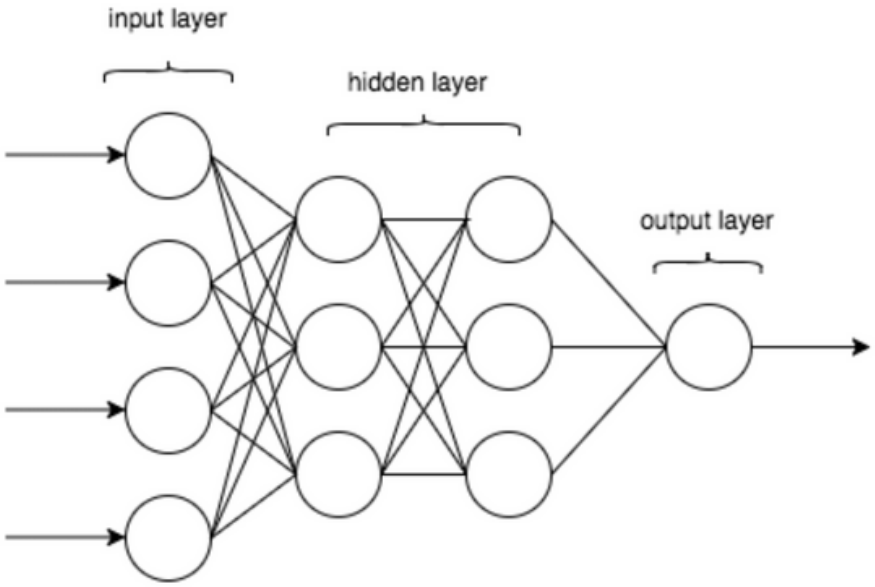

Fig 3.8.4 Dense Layers [21]

A dense layer is the most commonly used layer in neural networks and can learn complex patterns and relationships from the input data. A dense layer can also change the dimensionality of the output vector, which can be useful for tasks such as classification or regression.

In this project, we have used dense layers after some convolutional and pooling layers to create a fully connected network that can classify cow breeds. We used Dense layers in keras to implement dense layers with a given number of units and activation function.

3.11 Compile Model

Next, we need to compile our model. Compiling the model takes three parameters: optimizer, loss and metrics. The optimizer controls the learning rate. We will be using 'adam' as our optimizer. Adam is generally a good optimizer to use for many cases. The Adam optimizer adjusts the learning rate throughout training.

The learning rate determines how fast the optimal weights for the model are calculated. A smaller learning rate may lead to more accurate weights (up to a certain point), but the time it takes to compute the weights will be longer. We will use 'categorical_crossentropy' for our loss function. This is the most common choice for classification. A lower score indicates that the model is performing better. To make things even easier to interpret, we will use the 'accuracy' metric to see the accuracy score on the validation set when we train the model.

3.12 Perform Training

Now we will train our model. To train, we will use the 'fit()' function on our model with the following five parameters: training data (train_X), target data (train_y), validation split, the number of epochs and callbacks. The validation split will randomly split the data into use for training and testing.

During training, we will be able to see the validation loss, which gives the mean squared error of our model on the validation set. We will set the validation split at 0.2, which means that 20% of the training data we provide in the model will be set aside for testing model performance.

The number of epochs is the number of times the model will cycle through the data. The more epochs we run, the more the model will improve, up to a certain point. After that point, the model will stop improving during each epoch. In addition, the more epochs, the longer the model will take to run. To monitor this, we will use 'early stopping'.

3.13 Determine Accuracy and Try to Improve

Fig 3.9 Accuracy and Loss Function Graph

3.14 MACHINE LEARNING

Machine Learning is an integral part of artificial intelligence techniques that employ statistical and probabilistic methods to enable the machines to improve with experience. Any ML system consists of two components: data and algorithm. Data is what we allocate to the system in a form of input, while an algorithm is a method that works on the data and performs the analysis.

Machine learning is used for this project to create a system that can classify cow breeds from images. The system uses a convolutional neural network (CNN) model that is trained on a dataset of cow images labeled with their breeds. The CNN model can extract features from the images and learn to recognize the distinctive characteristics of each breed.

The system uses a website interface to allow users to upload their own cow images and get the breed predictions with the help of the trained model to predict the breed. Machine learning is used for this project because it can handle complex and high-dimensional data, such as images, and can learn from data without requiring explicit rules or human intervention.

Machine learning can also improve the accuracy and performance of the system by using more data and better algorithms. Machine learning can also enable the system to adapt to new data and situations, such as new breeds or variations in image quality.

3.14.1 Machine Learning Types

Machine learning can be broadly classified into three types based on the nature and availability of data and feedback:

- **Supervised learning:** The machine learning algorithm is given a set of labeled data, where each input is associated with a desired output or target. The algorithm learns a function that maps the inputs to the outputs and can use it to make predictions on new or unseen data. Examples of supervised learning tasks include classification, regression, anomaly detection and recommendation systems.
- **Unsupervised learning:** The machine learning algorithm is given a set of unlabeled data, where there is no predefined output or target. The algorithm learns to discover the hidden structure or patterns in the data and can use it to generate or summarize data. Examples of unsupervised learning tasks include clustering, dimensionality reduction, generative modeling and association rule mining.
- **Reinforcement learning:** The machine learning algorithm is given a goal or objective, and interacts with an environment by taking actions and receiving feedback in the form of rewards or penalties. The algorithm learns to optimize its behavior based on the feedback and can use it to achieve the goal or maximize the reward. Examples of reinforcement learning tasks include game playing, robotics, self-driving cars and resource management.

Machine learning has a wide range of applications in various domains and industries, such as:

- Healthcare: Machine learning can be used to diagnose diseases, analyze medical images, discover new drugs, personalize treatment plans and monitor patient health.

- Finance: Machine learning can be used to detect fraud, assess credit risk, optimize portfolio allocation, automate trading and provide financial advice.
- E-commerce: Machine learning can be used to recommend products, segment customers, predict demand, optimize pricing and enhance customer experience.
- Education: Machine learning can be used to grade assignments, provide feedback, tailor curriculum, assess student progress and facilitate online learning.
- Entertainment: Machine learning can be used to generate content, such as music, art, videos and games, as well as to analyze user preferences, personalize recommendations and improve user engagement.

3.14.2 Machine Learning Models and Algorithms

Machine learning models and algorithms are closely related concepts, but they are not the same. A machine learning algorithm is a set of rules or instructions that can be implemented in code and can process data to perform a specific task. A machine learning model is the output or result of applying a machine learning algorithm to a dataset.

A machine learning model consists of two components: the model data, which are the parameters or weights learned by the algorithm from the data, and the prediction algorithm, which is the function that uses the model data to make predictions on new or unseen data.

Different machine learning algorithms are suited to different goals, such as classification or prediction modeling, so data scientists use different algorithms as the basis for different models.

Machine learning algorithms are procedures that are implemented in code and are run on data to create machine learning models. Machine learning algorithms perform pattern recognition and learn from data, or are fit on a dataset.

There are many machine learning algorithms, such as linear regression, logistic regression, decision tree, artificial neural network, k-nearest neighbors, k-means, etc.

Machine learning algorithms can be categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning occurs when an algorithm is trained using labeled data, or data that is tagged with a label so that an algorithm can successfully learn from it. Unsupervised learning uses unlabeled data to train an algorithm and find patterns in the data itself. Reinforcement learning uses a mix of labeled and unlabeled data to train an algorithm that learns from its own actions and rewards.

There are around 50 ML algorithms utilized for solving issues in precision agriculture relating to different subcategories. All the utilized algorithms were grouped into nine different ML models.

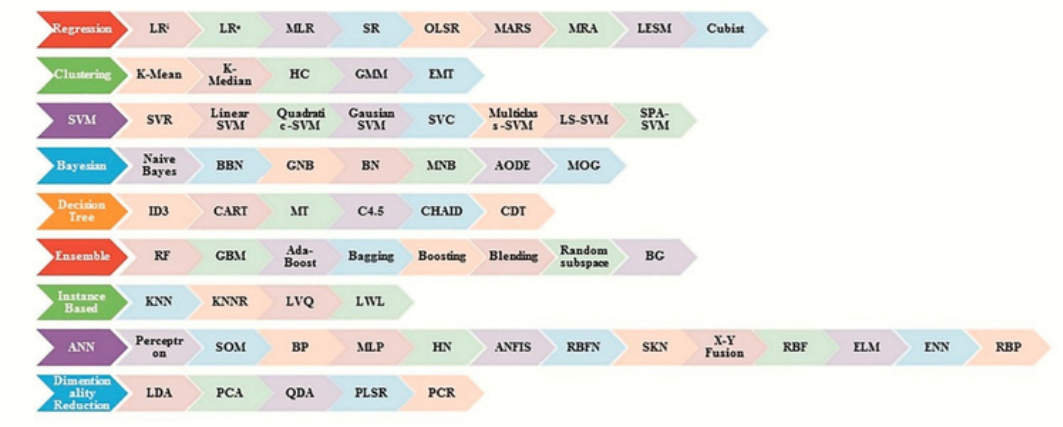

Fig 3.10 ML Models & Algorithms [19]

The machine learning model and algorithm used in this project are:

- **Model:** A convolutional neural network (CNN) model that can classify cow breeds from images. A CNN is a type of neural network that consists of multiple layers that can extract features from images and learn to recognize patterns. A CNN model can have different types of layers, such as convolutional layers, pooling layers, dropout layers, and dense layers. Each layer performs a specific function and transforms the input data into a more abstract representation.

- **Algorithm:** A supervised learning algorithm that can train the CNN model using a labeled dataset of cow images. The algorithm can use different techniques to optimize the performance and accuracy of the model, such as gradient descent, backpropagation, regularization, and data augmentation. The algorithm can also use different metrics to evaluate the model, such as accuracy, precision, recall, and F1-score.

3.14.2.1 Regression

Regression is a statistical method that can be used to estimate the relationship between a dependent variable and one or more independent variables. Regression can also be used to predict the value of the dependent variable based on the values of the independent variables.

Regression can be applied to different types of data and problems, such as:

- Continuous data: Regression can be used to model and predict continuous outcomes, such as sales, prices, or incomes. Examples of regression techniques for continuous data include linear regression, polynomial regression, and ridge regression.
- Categorical data: Regression can be used to model and predict categorical outcomes, such as yes/no, positive/negative, or red/green. Examples of regression techniques for categorical data include logistic regression, multinomial regression, and ordinal regression.
- Time series data: Regression can be used to model and predict outcomes that vary over time, such as stock prices, weather, or sales trends. Examples of regression techniques for time series data include autoregressive models, moving average models, and exponential smoothing models.

Regression can also be classified into different types based on the number and nature of the independent variables, such as:

- Simple regression: Regression that involves only one independent variable and one dependent variable.
- Multiple regression: Regression that involves more than one independent variable and one dependent variable.

- Linear regression: Regression that assumes a linear relationship between the independent variables and the dependent variable.
- Non-linear regression: Regression that does not assume a linear relationship between the independent variables and the dependent variable.

Regression is used in this project to predict the weight of cows based on their breed, age, and gender. The dependent variable is the weight of the cow, and the independent variables are the breed, age, and gender of the cow. The type of regression used in this project is multiple linear regression.

3.14.2.2 Clustering

Clustering is a type of unsupervised learning method that aims to group similar data points into clusters based on some measure of similarity or distance. Clustering can be used to discover the hidden structure or patterns in unlabeled data, and can use it to generate or summarize data.

Clustering can also be classified into different types based on the criteria and methods used to form the clusters, such as:

- Partitioning methods: These methods divide the data points into a predefined number of clusters by minimizing some criterion function that measures the distance or dissimilarity between the data points and the cluster centers.
- Hierarchical methods: These methods create a hierarchical structure of clusters by either merging smaller clusters into larger ones (agglomerative) or splitting larger clusters into smaller ones (divisive). The result is a tree-like representation called a dendrogram that shows the nested clusters at different levels.
- Density-based methods: These methods identify clusters as dense regions of data points that are separated by low-density regions. These methods can handle arbitrary shapes and sizes of clusters and are robust to noise and outliers. Examples of density-based methods include DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points to Identify Clustering Structure), and DENCLUE (Density-Based Clustering).

3.14.2.3 Support Vector Machines

SVM algorithms/models belong to a supervised learning category and are effectively applied for the two-class problem of classification. However, SVM can solve regression and clustering problems too. The capabilities of traditional SVM classifiers are being enhanced through multiple kernels functions (i.e., linear kernel, sigmoid kernel, non-linear kernel, RBF, Gaussian kernel, polynomial kernel, etc.).97 Some frequently used SVM algorithms are SVR, linear SVM, quadratic SVM, Gaussian SVM, SVC, multiclass SVM, least-square SVM, and successive projection algorithm SVM (SPA-SVM).

3.14.2.4 Bayesian

The bayesian is a probability-based supervised learning model, and capable to solve both classification and regression problems. NB, Gaussian naïve Bayes (GNB), Bayesian belief network (BBN), Bayesian network, multinomial naïve Bayes (MNB), averaged one dependence estimators (AODE) and the mixture of Gaussian are prominent algorithms of the BM.96 8 of 22 MAHMOOD ET AL.

3.14.2.5 Decision Tree

Decision Tree is a supervised learning model that resolves both classification and regression based problems. DT has a hierarchical structure and follows the divide and conquer technique to solve the problems. The commonly applied algorithms of the DT are- Iterative dichotomiser-3, CART, MT, C4.5, chi-squared automatic interaction detection (CHAID).

3.14.2.6 Ensemble Learning

Ensembles are a powerful and more advanced type of supervised learning model. Ensemble modulates the capabilities of multiple algorithms to improve the prediction outcomes. However; the Ensemble approach helps to generate a better prediction model compare to a single predictive model, but it is hard to understand the ensemble of multiple algorithms. Some well-known EM algorithms are- RF, GBM, AdaBoost, BG (bagging), boosting, stacked generalization (blending), random subspace, and boosting gradient.

3.14.2.7 Instance-based learning

Instance-based learning sometimes identified as memory-based or lazy learning; learns and builds the hypothesis from training data according to some similarity measures. Instead of estimating for entire instances, it uses only specific instances to perform regression or classification tasks. The instance-based model requires huge memory space to store data which leads to a complex structure. Some popular Instance-based algorithms are- KNN, learning vector quantization (LVQ), KNN-regression, and logically weighted learning (LWL).

3.14.2.8 Artificial Neural Network

ANN is a human brain-inspired information processing model. Similar to humans; ANN learns from examples and performs real-time tasks such as image processing, pattern recognition, classification of objects, natural language processing, and so forth.

ANN is a supervised type of learning model which usually solves classification and regression problems. ANN architecture consists of basically three layers; input, output, and one or multiple hidden layers. Data gets into the network through the input layer; learning is performed in hidden layers, and prediction results are obtained at output layers.

A vast number of ANN algorithms are reported in the literature such as perceptron, SOM, back-propagation (BP), MLP, Hopfield network (HN), ANFIS, RBF-network, SKN, X-Y fusion, RBF, ELM, ENN, resilient back-propagation (RBP), counter propagation, generalized regression neural network and self-adaptive evolutionary extreme learning machines (SAEELM).98 DL sometimes referred to as the deep neural network (DNN) is a subset of ML, resembles ANN. DL architecture consists of multiple layers (i.e., more than simple ANN) to perform complex processing of data and have capabilities to extract features on their own; and can be utilized to solve supervised, unsupervised, and semi-supervised learning problems. Now a day's DL architectures/models are frequently used by researchers and artificial intelligence experts due to their wide range of application scenarios.

3.14.2.9 Dimensionality Reduction

Dimensionality refers to features associated with the input data, and DR is a technique to reduce the features of the dataset. DR algorithms can be employed for all types of learning problems (i.e., supervised, unsupervised, and semi-supervised) and applied before regression, classification, or clustering models to reduce the dimensionality.

Commonly used DR algorithms are- LDA, quadratic discriminant analysis (QDA), PCA, partial least squares regression (PLSR), and principal component regression (PCR).

There are many different areas of agriculture where ML is being applied triumphantly. All the identified works related to ML application in agriculture are classified systematically into four broad categories viz. field condition management, crop management, species management, and live.

3.15 Website Implementation

3.15.1 HTML

HTML (Hyper Text Markup Language) is a fundamental language for building web pages and applications. It provides a structured way to organize and present content on the World Wide Web. HTML is composed of various elements that define the structure, layout, and functionality of web documents.

At its core, HTML is a markup language that uses tags to define elements within a web page. These tags are enclosed in angle brackets (< >) and consist of a tag name that describes the type of element and optional attributes that provide additional information about the element. The basic structure of an HTML document consists of an opening tag followed by and tags, and finally a closing tag.

HTML5 introduced new elements and features to enhance the capabilities of web pages. When combined with CSS, HTML enables developers to create visually appealing and interactive web experiences. HTML is a fundamental language for anyone involved in web development and plays a crucial role in the modern digital landscape.

The <head> section typically contains meta-information about the document, such as the document's title, character encoding, CSS stylesheets, and JavaScript scripts. It doesn't appear directly on the page but provides important instructions and references for the browser.

3.15.2 CSS

CSS (Cascading Style Sheets) is a powerful language used to define the visual appearance and layout of web documents written in HTML and XHTML. It allows developers to control the presentation aspects of a web page, such as colors, fonts, spacing, and positioning, separate from the underlying structure and content.

CSS works by targeting HTML elements and applying style rules to modify their appearance. These rules consist of selectors that specify which elements to target and declarations that define the desired styles. Selectors can be based on element names, classes, IDs, attributes, or even the relationship between elements (e.g., parent-child or sibling selectors).

CSS provides a wide range of properties to control various aspects of element styling. Some common properties include:

- Colors: CSS allows you to set the color of text and backgrounds using properties like "color" and "background-color". Colors can be defined using keywords, hexadecimal values, RGB or HSL values, or even color names.
- Fonts and Text: CSS provides properties like "font-family", "font-size", and "font weight" to control the appearance of text. It allows you to set the font family, size, weight, style, and other attributes. Additionally, you can control text alignment, spacing, decoration, and transformation using properties like "text-align", "line-height", "text decoration", and "text-transform".
- Layout and Positioning: CSS enables you to control the layout and positioning of elements on a web page. Properties like "display", "float", "position", and "flexbox" help you define how elements flow and interact with each other. With CSS, you can create responsive designs that adapt to different screen sizes and orientations.

- Box Model: The box model is a fundamental concept in CSS that defines how elements are rendered and spaced on the page. CSS properties like "margin", "padding", "border", and "box-sizing" allow you to control the size, spacing, and borders around elements, influencing their layout and appearance.
- Transitions and Animations: CSS provides properties like "transition" and "@keyframes" that enable the creation of smooth transitions and animations. With CSS animations, you can bring elements to life by defining keyframes and specifying their timing, duration, and easing functions.
- Responsive Design: CSS plays a vital role in creating responsive web designs that adapt to different devices and screen sizes. Media queries allow you to apply specific styles based on the device's characteristics, such as screen width, height, orientation, and resolution. This enables the creation of mobile-friendly and user-friendly experiences.

3.15.3 JavaScript

JavaScript is a dynamic programming language primarily used for adding interactivity and functionality to web pages. It allows developers to create interactive elements, handle events, manipulate and validate data, and build complex applications that run on the client side.

The evolution of JavaScript, a programming language commonly used for building web applications, has seen a number of significant changes and improvements since it was first introduced in the mid-1990s. The first version of JavaScript, known as ECMAScript 1, was released in 1997, and provided basic language features such as control structures, data types, and error handling.

Since then, JavaScript has undergone several major revisions, with the release of ECMAScript 3 in 1999, ECMAScript 5 in 2009, and ECMAScript 6 (also known as ECMAScript 2015) in 2015. These updates introduced a range of new features and improvements, such as improved support for object-oriented programming, regular expressions, and new data types.

The most recent version of JavaScript is ECMAScript 2020, which was released in 2020. This update introduced new features such as optional chaining and nullish coalescing, as well as improvements to existing features such as regular expressions and string manipulation.

JavaScript can be embedded directly within HTML documents using the tag or included in external JavaScript files that are linked to HTML pages. The language provides a wide range of features and functionalities, including variables, data types, operators, control structures, functions, objects, and more.

JavaScript provides powerful features for manipulating and traversing the Document Object Model (DOM), which represents the structure of an HTML document. Developers can use JavaScript to dynamically create, modify, or remove elements on the page, change their styles, and respond to user interactions.

Another notable aspect of JavaScript is its support for asynchronous programming through callbacks, promises, and async/await syntax. Asynchronous programming allows for non blocking operations, ensuring that long-running tasks like network requests or file operations don't block the user interface. This enhances performance and responsiveness.

3.15.4 Python

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Created by Guido van Rossum and first released in 1991. Python is a popular programming language that can be used for various purposes, such as data analysis, web development, machine learning, and more. Python is known for its simplicity, readability, and versatility.

The evolution of Python, a popular programming language, has seen a number of significant changes and improvements since it was first released in 1991. The earliest versions of Python were relatively basic, and focused on providing a simple and easy-to-use language for writing programs.

Since then, Python has undergone several major revisions, with the release of Python 2.0 in 2000 and Python 3.0 in 2008. These updates introduced a range of new features and improvements, such as improved support for object-oriented programming, new data types, and enhanced error handling.

The most recent version of Python is Python 3.9, which was released in October 2020. This update introduced a number of new features and improvements, such as support for pattern matching, a new debugging tool, and improved performance.

Overall, the evolution of Python has been driven by the need to provide programmers with a powerful and versatile language for building a wide range of applications. As the field of programming continues to evolve and become more complex, it is likely that Python will continue to evolve and improve in order to keep pace with these changes.

Python is commonly used in scientific computing and data analysis due to libraries like NumPy, Pandas, and Matplotlib. These libraries provide powerful tools for numerical computations, data manipulation, and visualization, making Python a popular choice for researchers and data scientists.

Python's popularity has also surged in machine learning and artificial intelligence. Libraries such as TensorFlow, PyTorch, and scikit-learn provide powerful tools for developing machine learning models and performing deep learning tasks. Python's simplicity and ease of use, combined with the availability of these libraries, have contributed to its dominance in this rapidly evolving field.

In this project, Python is used to implement the machine learning model and algorithm for cow breed classification. Also used to create the website interface that allows users to upload their own cow images and get the breed predictions. Some of the Python libraries and frameworks that can be used in this project are:

- Keras: A high-level neural network API that can be used to build and train deep learning models using TensorFlow as the backend.
- TensorFlow: An open-source platform that provides various tools and resources for machine learning and deep learning.
- Scikit-learn: A library that provides various tools and algorithms for machine learning, such as data preprocessing, model selection, evaluation, and clustering.
- Flask: A lightweight web framework that can be used to create web applications using Python.
- Pillow: A library that provides various tools and functions for image processing and manipulation using Python.

3.15.5 Django Framework

Django is a web framework that is written in Python and follows the model-template-views (MTV) architectural pattern. Django is designed to facilitate rapid and clean web development, and provides various features and tools, such as:

- An object-relational mapper (ORM) that allows you to define and manipulate data models using Python classes and query the database using a high-level API.
- A URL dispatcher that maps URLs to views, which are Python functions or classes that handle HTTP requests and generate HTTP responses.
- A template system that allows you to create dynamic HTML pages using a syntax that mixes HTML and Python code.
- A built-in admin interface that allows you to manage your data models and perform CRUD (create, read, update, delete) operations using a web browser.
- A caching framework that can use various cache methods to improve the performance and scalability of your web applications.

Django is used in this project to create the website interface that allows users to upload their own cow images and get the breed predictions. Django is also used to integrate the machine learning model and algorithm for cow breed classification with the web application. Some of the steps involved in using Django for this project are:

- Setting up a Django development environment by installing Python, Django, and other dependencies.
- Creating a Django project and app by using the django-admin utility.
- Defining the data models for the cow images and breeds by using the Django ORM.
- Registering the data models with the Django admin interface by using the admin.py file.
- Creating the views for handling the user requests and generating the responses by using the views.py file.

- Mapping the URLs to the views by using the urls.py file.
- Creating the templates for rendering the HTML pages by using the template system.
- Loading and using the machine learning model and algorithm for cow breed classification by importing them in the views.py file.

3.15.6 Django REST Framework

Django REST framework is a library that extends the Django web framework and provides various tools and features for building RESTful APIs. RESTful APIs are web services that follow the REST (Representational State Transfer) principles and allow data exchange between different applications or systems using standard HTTP methods and formats.

Django REST framework used to create APIs for cow breeds in this project by following these steps:

- Install Django REST framework by using pip or adding it to the requirements.txt file.
- Add 'rest_framework' to the INSTALLED_APPS setting in the settings.py module.
- Create a serializer for the cow breed model by using the rest_framework.serializers.ModelSerializer class and specifying the fields to be included in the API response.
- Create a view for handling the API requests and generating the API responses by using the rest_framework.decorators.api_view decorator and the rest_framework.response.Response class.
- Map the URL to the view by using the urls.py module and the rest_framework.urls.include function.

3.15.7 API

An Application Programming Interface (API) is a set of rules and protocols that allows different software applications to communicate and interact with each other. APIs define how different components of software systems should interact, enabling developers to access and use the functionality of other applications, services, or platforms.

APIs can be categorized into different types based on their purpose and usage. Some common types of APIs include:

- Web APIs: These are APIs specifically designed for web applications. Web APIs are typically exposed over HTTP (Hypertext Transfer Protocol) and enable interaction with remote services or retrieve data from servers. Web APIs are widely used for building web applications, mobile apps, and integrating different systems.
- REST APIs: Representational State Transfer (REST) is an architectural style for designing networked applications. REST APIs are based on a set of principles and constraints that define how resources are represented and accessed using standard HTTP methods such as GET, POST, PUT, and DELETE. REST APIs are stateless, scalable, and widely used for building web services.
- SOAP APIs: Simple Object Access Protocol (SOAP) is a protocol for exchanging structured information in web services using XML (eXtensible Markup Language). SOAP APIs provide a formal contract for interaction and use XML messages for communication. They are more rigid and heavyweight compared to REST APIs but are still used in enterprise systems.
- Library or Framework APIs: These APIs are specific to programming languages, frameworks, or libraries. They provide a set of functions, classes, or modules that developers can use to access pre-built functionality and services. For example, the Python programming language has various built-in APIs and libraries like the Python Standard Library and third-party libraries like NumPy, Django, and TensorFlow.

3.15.8 SQLITE3 DATABASE

SQLite3 is a lightweight and self-contained database engine that can store data in a single file. It is the default database for Django projects and can be used for development and testing purposes.

One of the key features of SQLite3 is its serverless architecture. Unlike traditional client server databases, SQLite3 runs as a library linked directly into the application. This means that the database engine and the application code operate in the same address space, eliminating the need for a separate database server.

SQLite3 databases are self-contained and stored in a single file, making them highly portable. The entire database, including tables, indexes, views, and triggers, is stored within a single file on the file system. This simplicity allows developers to easily distribute and share SQLite3 databases as standalone files.

Despite its small footprint, SQLite3 supports a wide range of SQL features. It supports standard SQL syntax and provides ACID (Atomicity, Consistency, Isolation, Durability) transactions, ensuring data integrity. It also supports various data types, including integers, floating-point numbers, strings, and date/time values. SQLite3 provides a rich set of SQL commands for creating, modifying, and querying databases, making it a powerful tool for managing data.

3.16 Integrate ML Model with Website

After finishing the website we will need integrate the ML model that we have build with the website. The requirements and dependencies to run the model are also added to development environment with the requirements of the website.

Before running the website, we made some changes in the settings.py file of Django project like providing path of the model, path of dataset and allowed host.

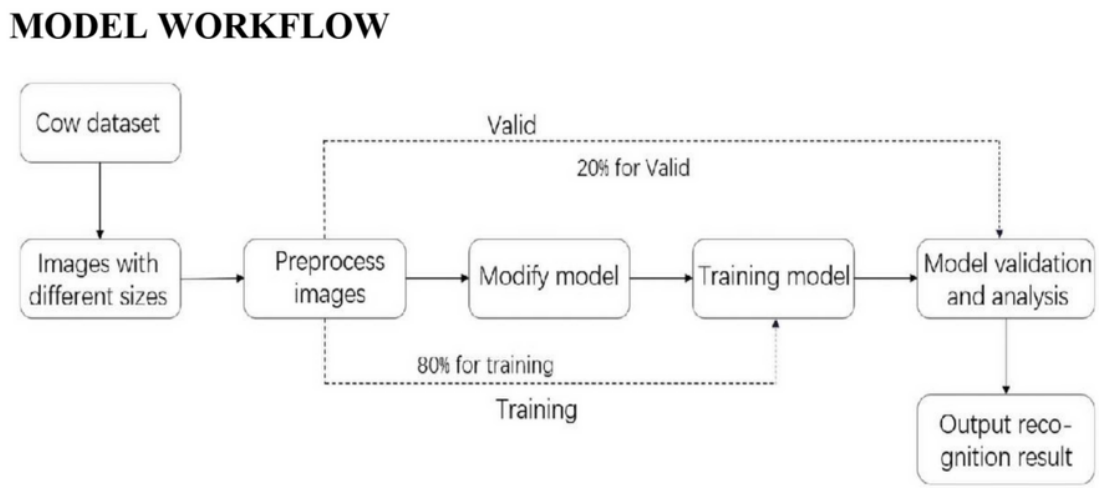

Fig 3.11 Model Workflow

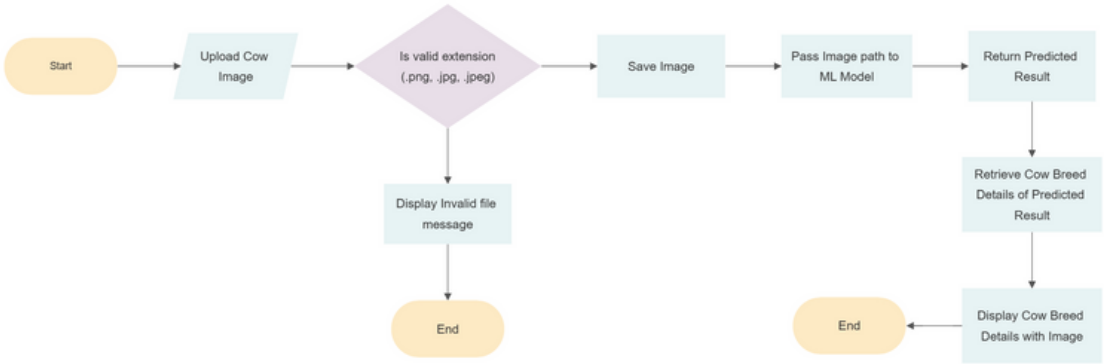

Fig 3.12 Website Workflow on Image Upload

CHAPTER 4
RESULTS AND DISCUSSION

We have trained the classification model multiple times in order to increase the accuracy of the model. Inorder to get the best results we have trained the model multiple times by changing the number of augmentations and also the number of epochs. Analyzing the various results achieved inorder to enhance the further results is an important part of work. Apart from enhancing the model working in the backend we have also worked on improving the UI of the Web App.

The results achieved were as follows:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 8, 8, 2048)        21802784

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dropout (Dropout)           (None, 2048)              0

 dense (Dense)               (None, 512)               1049088

 dense_1 (Dense)             (None, 512)               262656

 dense_2 (Dense)             (None, 4)                 2052

=================================================================
Total params: 23,116,580
Trainable params: 1,313,796
Non-trainable params: 21,802,784
```

Fig 4.1 Initialized Inception V3 Model

	Target_Labels	Predictions
0	Red Sindhi	Red Sindhi
1	Rathi	Rathi
2	Gir	Sahiwal
3	Red Sindhi	Sahiwal
4	Rathi	Sahiwal
...	...	...
195	Sahiwal	Sahiwal
196	Rathi	Rathi
197	Sahiwal	Rathi
198	Gir	Gir
199	Sahiwal	Sahiwal

200 rows × 2 columns

Fig 4.2 Target Label and Prediction Table

Accuracy of Model

```
correct = (target_labels == predictions)
accuracy = correct.sum()/correct.size
print(accuracy)
```

The Accuracy Achieved:

```
0.738255033557047
```

Fig 4.3 Model Accuracy

```
  if __name__ == '__main__':
Epoch 1/40
72/72 [==============================] - 67s 740ms/step - loss: 1.3097 - accuracy: 0.3750 - val_loss: 1.1719 - val_accuracy: 0.4965
Epoch 2/40
72/72 [==============================] - 48s 663ms/step - loss: 1.1814 - accuracy: 0.4696 - val_loss: 1.0908 - val_accuracy: 0.5243
Epoch 3/40
72/72 [==============================] - 47s 646ms/step - loss: 1.0973 - accuracy: 0.5139 - val_loss: 1.0331 - val_accuracy: 0.5903
Epoch 4/40
72/72 [==============================] - 48s 668ms/step - loss: 1.0560 - accuracy: 0.5512 - val_loss: 1.0093 - val_accuracy: 0.5851
Epoch 5/40
72/72 [==============================] - 47s 650ms/step - loss: 1.0194 - accuracy: 0.5699 - val_loss: 0.9714 - val_accuracy: 0.5990
Epoch 6/40
72/72 [==============================] - 47s 653ms/step - loss: 0.9882 - accuracy: 0.5920 - val_loss: 0.9470 - val_accuracy: 0.6233
Epoch 7/40
72/72 [==============================] - 46s 632ms/step - loss: 0.9964 - accuracy: 0.5781 - val_loss: 1.0012 - val_accuracy: 0.5712
Epoch 8/40
72/72 [==============================] - 45s 630ms/step - loss: 0.9517 - accuracy: 0.6089 - val_loss: 0.9207 - val_accuracy: 0.6319
Epoch 9/40
72/72 [==============================] - 49s 678ms/step - loss: 0.9169 - accuracy: 0.6163 - val_loss: 0.9129 - val_accuracy: 0.6163
Epoch 10/40
72/72 [==============================] - 47s 654ms/step - loss: 0.9409 - accuracy: 0.6076 - val_loss: 0.8967 - val_accuracy: 0.6476
Epoch 11/40
72/72 [==============================] - 48s 665ms/step - loss: 0.9070 - accuracy: 0.6293 - val_loss: 0.8532 - val_accuracy: 0.6632
Epoch 12/40
72/72 [==============================] - 47s 656ms/step - loss: 0.8923 - accuracy: 0.6372 - val_loss: 0.8686 - val_accuracy: 0.6441
Epoch 13/40
72/72 [==============================] - 49s 677ms/step - loss: 0.8875 - accuracy: 0.6233 - val_loss: 0.8517 - val_accuracy: 0.6441
Epoch 14/40
72/72 [==============================] - 46s 641ms/step - loss: 0.8688 - accuracy: 0.6419 - val_loss: 0.8364 - val_accuracy: 0.6354
Epoch 15/40
72/72 [==============================] - 47s 658ms/step - loss: 0.8865 - accuracy: 0.6398 - val_loss: 0.8169 - val_accuracy: 0.6753
Epoch 16/40
72/72 [==============================] - 48s 661ms/step - loss: 0.8462 - accuracy: 0.6506 - val_loss: 0.8183 - val_accuracy: 0.6510
Epoch 17/40
72/72 [==============================] - 49s 681ms/step - loss: 0.8288 - accuracy: 0.6719 - val_loss: 0.7921 - val_accuracy: 0.6840
Epoch 18/40
72/72 [==============================] - 47s 654ms/step - loss: 0.8447 - accuracy: 0.6523 - val_loss: 0.7929 - val_accuracy: 0.6823
Epoch 19/40
72/72 [==============================] - 46s 633ms/step - loss: 0.8439 - accuracy: 0.6580 - val_loss: 0.7763 - val_accuracy: 0.6979
Epoch 20/40
72/72 [==============================] - 48s 659ms/step - loss: 0.8349 - accuracy: 0.6697 - val_loss: 0.7816 - val_accuracy: 0.6632
Epoch 21/40
72/72 [==============================] - 49s 682ms/step - loss: 0.8212 - accuracy: 0.6645 - val_loss: 0.8802 - val_accuracy: 0.6146
Epoch 22/40
```

Fig 4.4.1 Accuracy while Training Model

```
Epoch 19/40
72/72 [==============================] - 46s 633ms/step - loss: 0.8439 - accuracy: 0.6580 - val_loss: 0.7763 - val_accuracy: 0.6979
Epoch 20/40
72/72 [==============================] - 48s 659ms/step - loss: 0.8349 - accuracy: 0.6697 - val_loss: 0.7816 - val_accuracy: 0.6632
Epoch 21/40
72/72 [==============================] - 49s 682ms/step - loss: 0.8212 - accuracy: 0.6645 - val_loss: 0.8802 - val_accuracy: 0.6146
Epoch 22/40
72/72 [==============================] - 48s 662ms/step - loss: 0.8093 - accuracy: 0.6766 - val_loss: 0.7713 - val_accuracy: 0.6840
Epoch 23/40
72/72 [==============================] - 47s 655ms/step - loss: 0.8037 - accuracy: 0.6732 - val_loss: 0.7648 - val_accuracy: 0.6823
Epoch 24/40
72/72 [==============================] - 45s 628ms/step - loss: 0.7936 - accuracy: 0.6688 - val_loss: 0.7372 - val_accuracy: 0.7031
Epoch 25/40
72/72 [==============================] - 47s 652ms/step - loss: 0.7977 - accuracy: 0.6806 - val_loss: 0.7704 - val_accuracy: 0.6736
Epoch 26/40
72/72 [==============================] - 47s 652ms/step - loss: 0.7788 - accuracy: 0.6940 - val_loss: 0.7472 - val_accuracy: 0.7031
Epoch 27/40
72/72 [==============================] - 46s 641ms/step - loss: 0.7610 - accuracy: 0.6962 - val_loss: 0.7093 - val_accuracy: 0.7083
Epoch 28/40
72/72 [==============================] - 45s 627ms/step - loss: 0.7431 - accuracy: 0.7070 - val_loss: 0.7239 - val_accuracy: 0.7101
Epoch 29/40
72/72 [==============================] - 48s 664ms/step - loss: 0.7618 - accuracy: 0.6845 - val_loss: 0.7152 - val_accuracy: 0.7014
Epoch 30/40
72/72 [==============================] - 47s 649ms/step - loss: 0.7445 - accuracy: 0.7044 - val_loss: 0.6937 - val_accuracy: 0.7205
Epoch 31/40
72/72 [==============================] - 47s 650ms/step - loss: 0.7437 - accuracy: 0.7001 - val_loss: 0.6954 - val_accuracy: 0.7153
Epoch 32/40
72/72 [==============================] - 45s 625ms/step - loss: 0.7535 - accuracy: 0.7088 - val_loss: 0.7246 - val_accuracy: 0.6788
Epoch 33/40
72/72 [==============================] - 48s 665ms/step - loss: 0.7330 - accuracy: 0.7144 - val_loss: 0.6829 - val_accuracy: 0.7274
Epoch 34/40
72/72 [==============================] - 47s 653ms/step - loss: 0.7076 - accuracy: 0.7196 - val_loss: 0.6744 - val_accuracy: 0.7274
Epoch 35/40
72/72 [==============================] - 46s 645ms/step - loss: 0.7071 - accuracy: 0.7209 - val_loss: 0.6823 - val_accuracy: 0.7135
Epoch 36/40
72/72 [==============================] - 46s 644ms/step - loss: 0.7309 - accuracy: 0.7079 - val_loss: 0.6906 - val_accuracy: 0.7205
Epoch 37/40
72/72 [==============================] - 45s 621ms/step - loss: 0.7129 - accuracy: 0.7166 - val_loss: 0.6706 - val_accuracy: 0.7483
Epoch 38/40
72/72 [==============================] - 46s 635ms/step - loss: 0.7406 - accuracy: 0.7083 - val_loss: 0.6994 - val_accuracy: 0.7188
Epoch 39/40
72/72 [==============================] - 45s 630ms/step - loss: 0.6955 - accuracy: 0.7257 - val_loss: 0.6494 - val_accuracy: 0.7535
Epoch 40/40
72/72 [==============================] - 45s 628ms/step - loss: 0.6835 - accuracy: 0.7179 - val_loss: 0.6718 - val_accuracy: 0.7413
```

Fig 4.4.2 Accuracy at Lat Epochs

Fig 4.5.1 Predicting Images along Their Confidence Level

Fig 4.5.2 Predicting Images along Thier Confidence Level

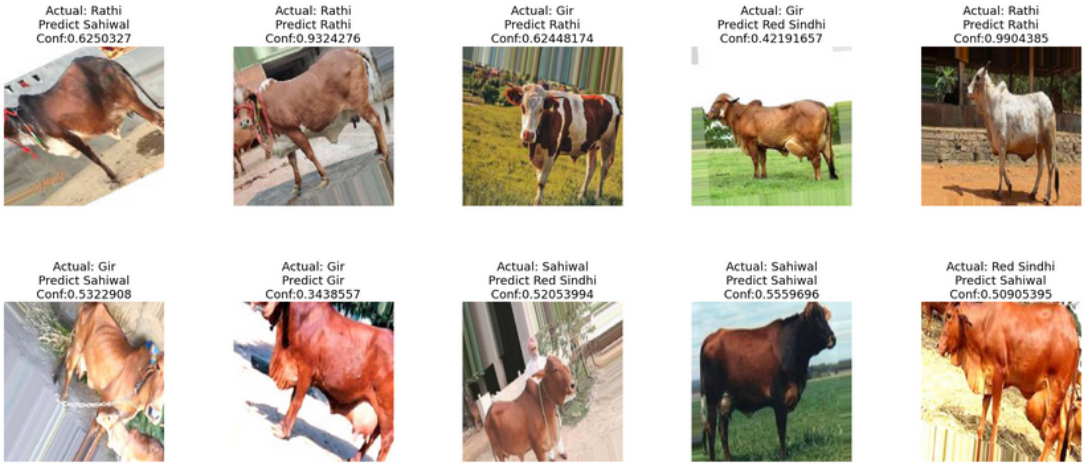

Fig 4.5.3 Predicting Images along Their Confidence Level

Reset Filters Records: 9 Search 9 records...

	id	name	photo	description	key_information
	Search column...	Search column...	Search column...	Search column...	Search column...
1	1	red sindhi	picture/red_sindhi.jpg	Red sindhi is originat...	The price for this cow...
2	2	sahiwal	picture/sahiwal.jpg	It is a breed of zebu c...	It produces about 8 t...
3	3	gir	picture/gir.jpg	It is a breed of zebu c...	Height 130 cm for fe...
4	4	hariana	picture/hariana.jpg	Hariana is an Indian c...	Height for males is b...
5	5	kankrej	picture/kankrej.jpg	It is a breed of zebui...	Weight of 550 - 570 ...
6	6	rathi	picture/rathi.jpg	Rathi is a breed of ca...	Weight of bulls is 350...
7	7	deoni	picture/deoni.jpg	It is a native cow bre...	Average height rang...
8	8	tharparkar	picture/tharparkar.jpg	Tharparkar is a breed...	Average height of co...
9	9	krishna_valley	picture/krishna_valle...	Krishna Valley is a br...	The average height o...

Fig 4.6 Cow Breed Database Table

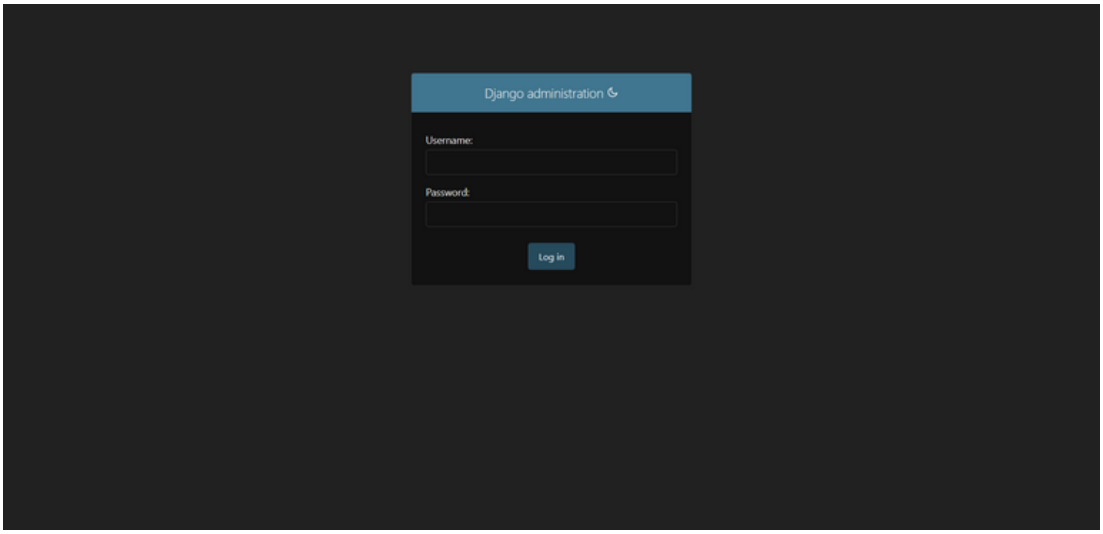

Fig 4.7 Login Page Screenshot

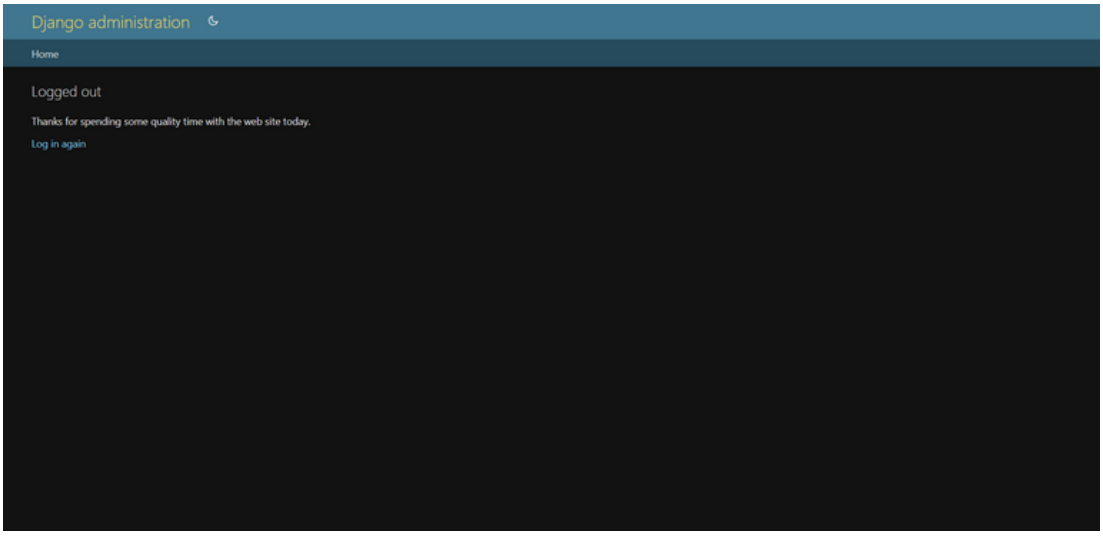

Fig 4.8 Logout Page Screenshot

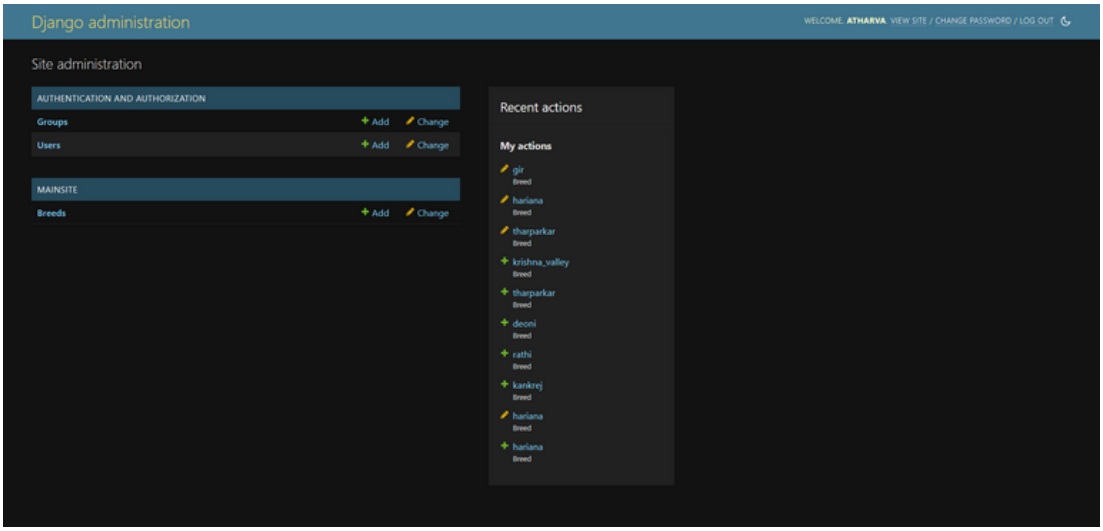

Fig 4.9 Admin Page Screenshot

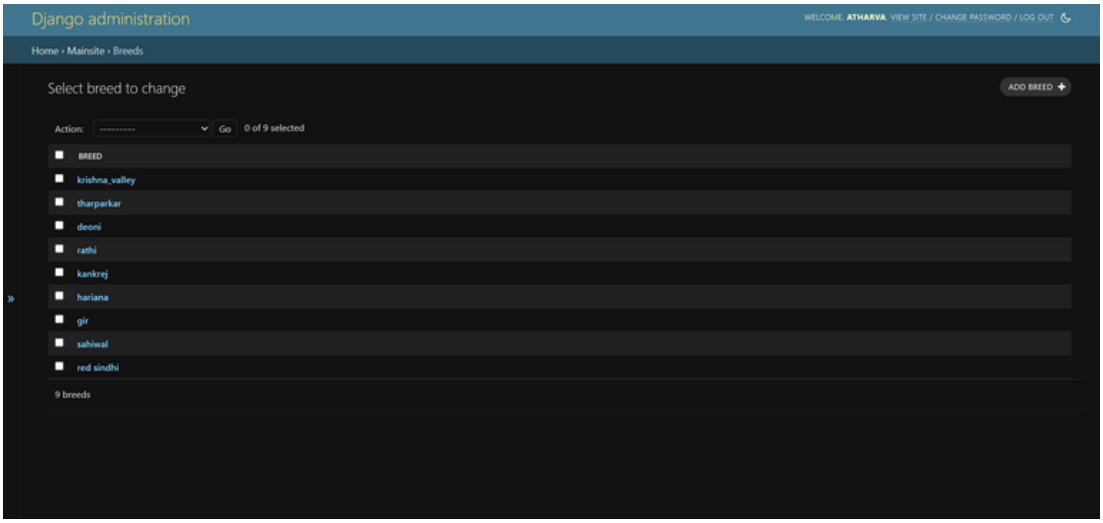

Fig 4.10 Admin Cow Breeds Page Screenshot

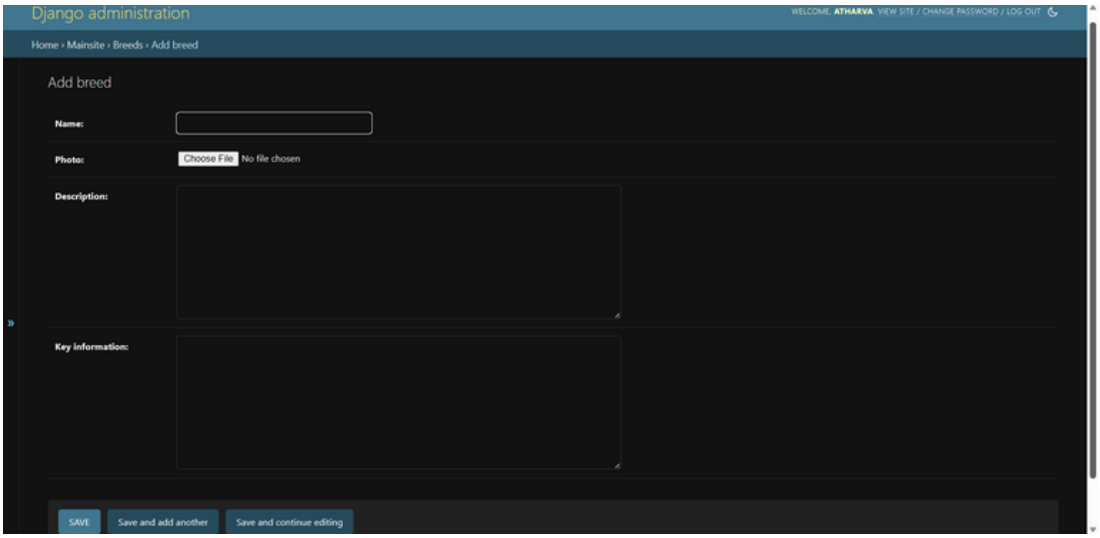

Fig 4.11 Add Cow Breed Page Screenshot

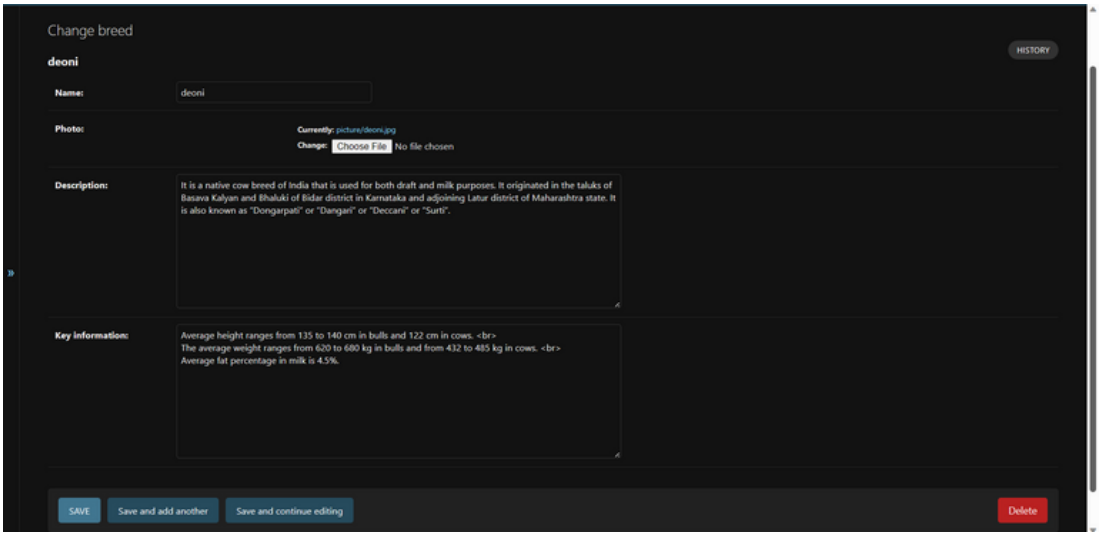

Fig 4.12 Edit Breed Page Screenshot

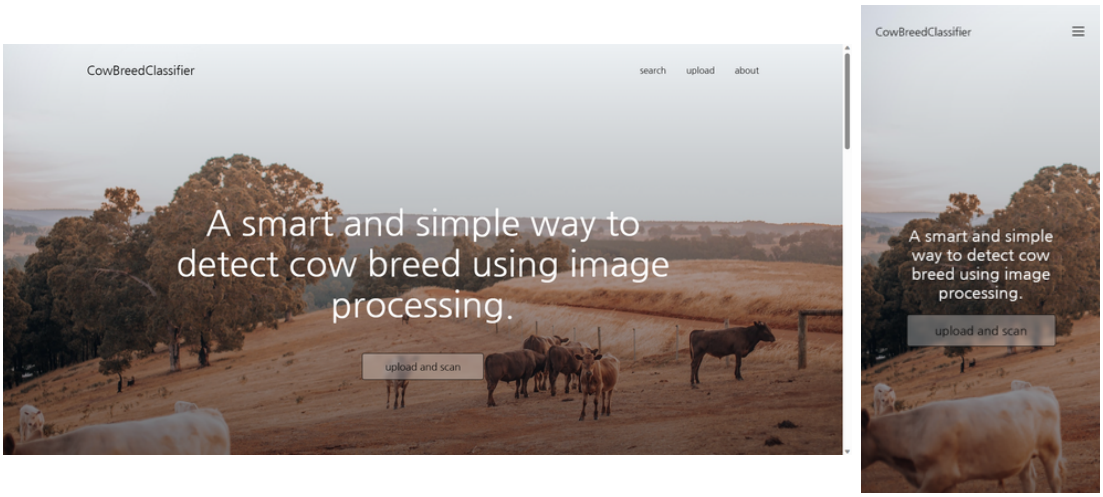

Fig 4.13.1 Home page Screenshots

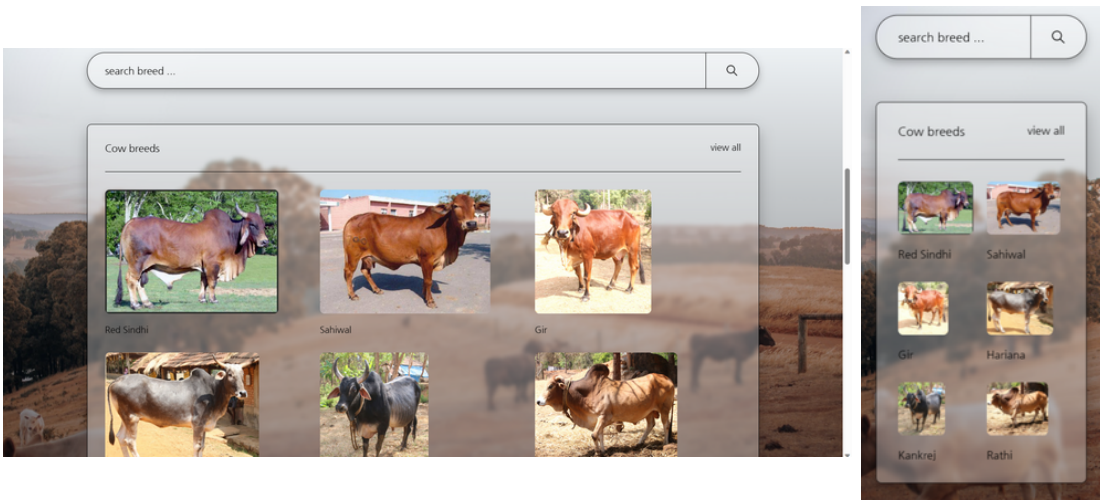

Fig 4.13.2 Home page (listed cow breeds) Screenshots

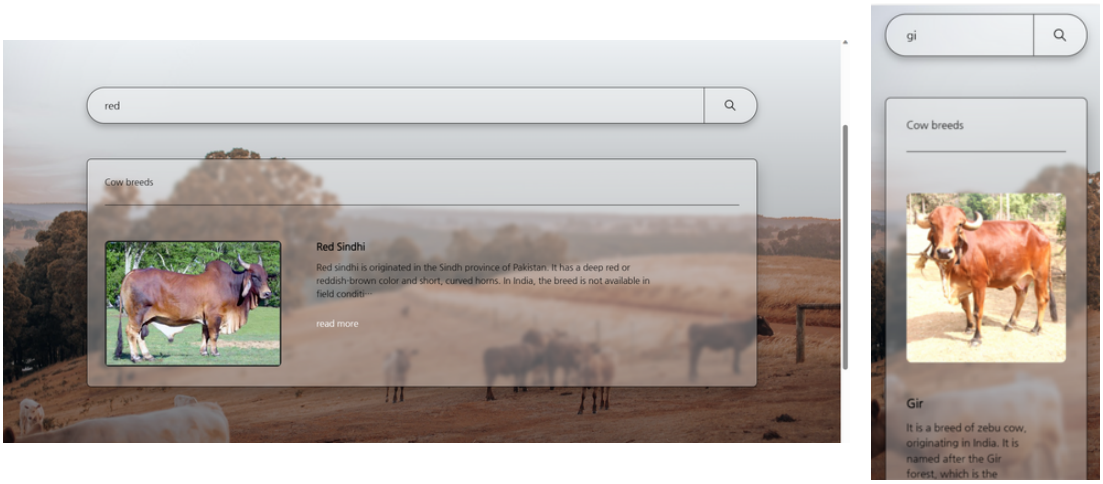

Fig 4.14 Search page Screenshots

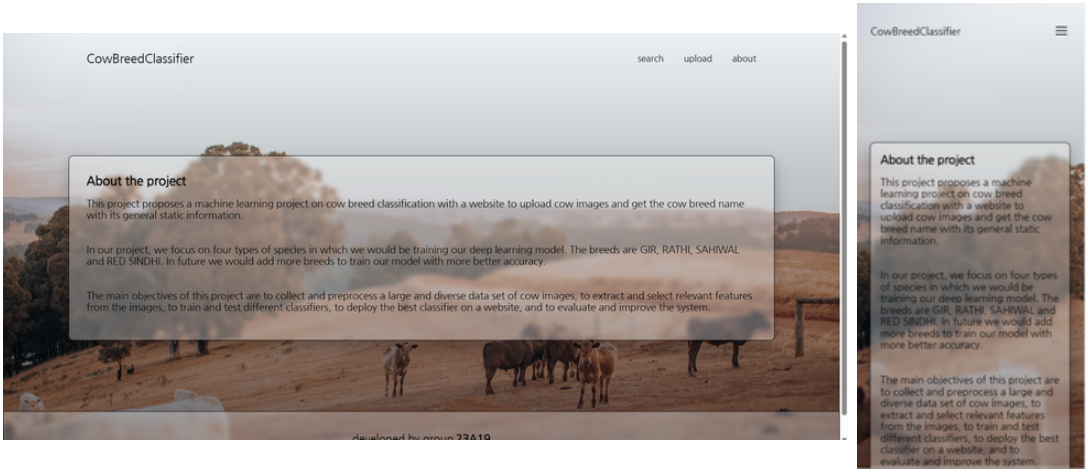

Fig 4.15 About Page Screenshots

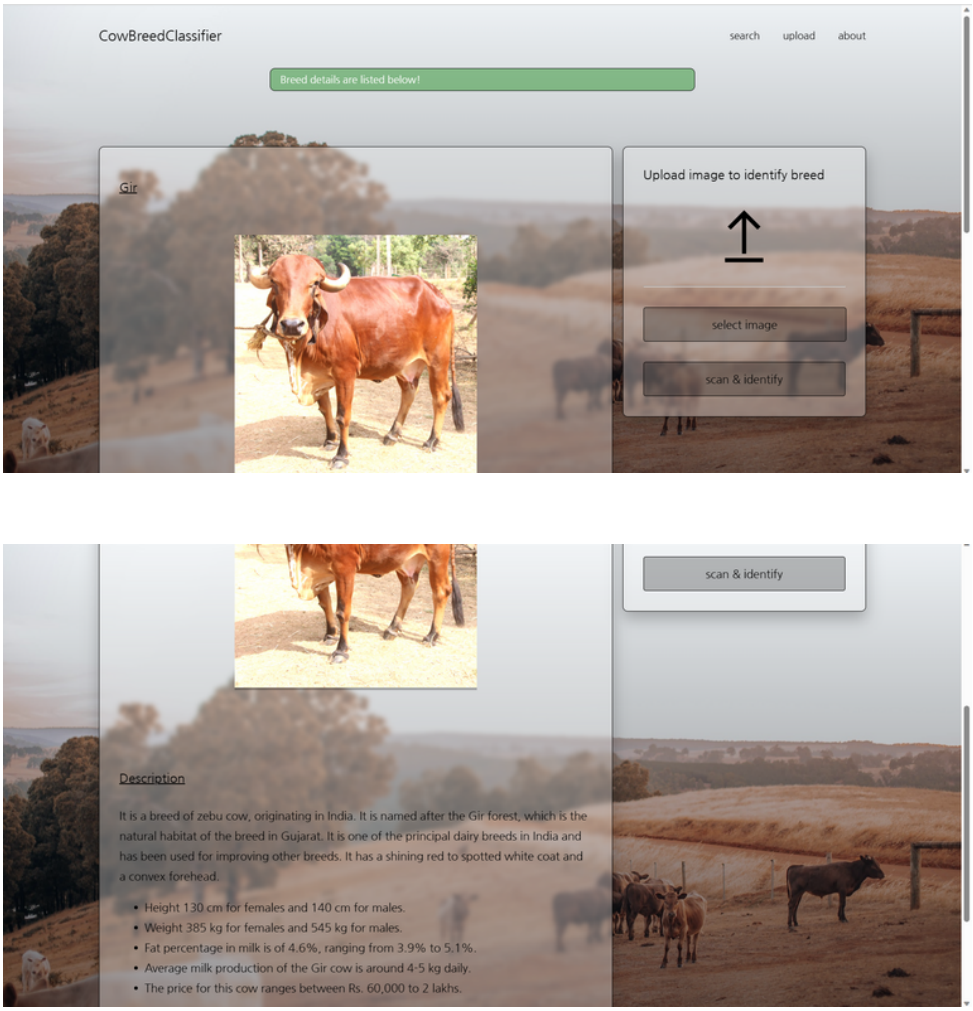

Fig 4.16 Upload and Detail Page Screenshots

4.1 Merits

1. It can help in preserving and promoting the genetic diversity of cow breeds in India and other countries.
2. It can provide useful information for farmers, researchers and policy makers on the characteristics and performance of different cow breeds.
3. It can facilitate the selection and improvement of cow breeds for different production systems and environments.
4. It can demonstrate the application of computer vision and machine learning techniques to a real-world problem.
5. It can showcase the capabilities of SQLite3 database and Django framework for building a simple and efficient website.

4.2 Demerits

1. It may not be able to capture all the nuances and variations of cow breeds that may depend on other factors besides phenotypic features.
2. It may require a lot of time and resources to collect, preprocess and label a large and diverse dataset of cow images.
3. It may face technical challenges in implementing and deploying the classifier and the website using SQLite3 database and Django framework.
4. It may have to deal with ethical and legal issues related to the collection, storage and use of cow images and data.

4.3 Limitations

1. It may require a large and diverse dataset of cow images to train and test the classifier with high accuracy and generalization. Currently it is able classify 4 cow breeds.
2. It may face difficulties in dealing with noisy, low-quality or occluded images that can affect the feature extraction and classification process.
3. It may encounter variations in the phenotypic features of cows due to environmental factors, crossbreeding or mutations that can cause misclassification or confusion.
4. It may have to deal with the scalability and security issues of SQLite3 database and Django framework when deploying the website to a larger audience or a public domain.

4.4 Future Scope

1. It can be extended to include more cow breeds from different regions and countries and compare their features and performance.
2. It can be improved to use more advanced and robust techniques for image processing, feature extraction and classification, such as deep neural networks, convolutional neural networks or transfer learning.
3. It can be integrated with other sources of data and information, such as genomic, phenotypic or production data, to provide a comprehensive and holistic analysis of cow breeds.
4. It can be adapted to other domains or applications, such as animal identification, tracking, health monitoring or disease diagnosis.

CONCLUSION

The main aim of this model is to learn how to use a machine learning classification tool to classify images, namely cow breeds. The application is properly proved with all sorts of cow images which gives faithful and precise results. A convolution neural network is a learning method for data analysis and predictions. Nowadays it has become a very popular image classification problem.

We aimed to improve the accuracy, stability and speed of cow identification. We explored new methods of cow identification with strong practical application capabilities to promote the development of intelligent cattle farming. In this study, we proposed a method for extracting multiscale hierarchical features for cow identification based on a lightweight convolutional neural network.

Using a deep learning convolutional neural network, the large network, Alex net, was used as a skeleton network and was improved by adding a multiscale extraction module. We introduced the short-circuit connection Basic Block and combined it with the SE attention mechanism to build a lightweight convolutional neural network model to train and recognize the side-view images of cows, achieving a final recognition rate of 97.95%.

The results show that the model is suitable for identifying cows in complex environments. Our lightweight, high-precision model for identifying individual cows has potential for practical application. Among the models used for cow identification, the proposed model has a small number of parameters; however, there is still room for improvement in recognition accuracy.

In future work, while ensuring that the model remains lightweight, we will improve its recognition accuracy. This will provide technical support for individual cow identification in complex environments and provide a scientific basis for intelligent cow breeding management.

REFERENCES

[1] Aristoteles. *Historia Animalium*; Cambridge University Press: Cambridge, UK, 2002.

[2] Lenstra, J.A.; Groeneveld, L.F.; Eding, H.; Kantanen, J.; Williams, J.L.; Taberlet, P.; Nicolazzi, E.L.; Sölkner, J.; Simianer, H.; Ciani, E. *et al.* Molecular tools and analytical approaches for the characterization of farm animal diversity. *Anim. Genet.* 2011, in press.

[3] Mommens, G. Boviene Microsatallieten Als DNA-Merkers; Een Studie Naar De Genetische Diversiteit in Runder Populaties; Universiteit Gent: Gent, Belgium, 2000.

[4] Martin-Burriel, I.; Rodellar, C.; Lenstra, J.A.; Sanz, A.; Cons, C.; Osta, R.; Reta, M.; De Arguello, S.; Sanz, A.; Zaragoza, P. Genetic diversity and relationships of endangered Spanish cattle breeds. *J. Hered.* 2007, *98*, 687-691.

[5] Felius, M. *Genus Bos: Cattle Breeds of the World. MSD-AGVET*; Merck and Co.: Rahway, NJ, USA, 1985.

[6] McQueen RJ, Garner SR, Nevill-Manning CG, Witten IH. Applying machine learning to agricultural data. Comput Electron Agric. 1995.

[7] Cow Breed compare image source(https://sunrisea2milk.com/Gir-Cows-identity)

[8] Gir, Rathi, Red Sindhi, Sahiwal (https://www.naturalfarmerskerala.com/gir-cow-indian-cows-best-milch-cow/,https://alchetron.com/cdn/sahiwal-cattle-54ebbca3-a77f-46cf-ab9c-9db186094c1-resize-750.jpeg, https://4.bp.blogspot.com/-_bpciNjj2KQ/WAjR9G9j3kI/AAAAAAAACq0/F8ct6Jat4uMlfGgZMYSkW2iLceoEFz4wACLcB/s600/Rathi%2BCattle.jpg,https://th.bing.com/th/id/OIP.uUjGEwNsGWoYyjOeRjBkZgAAAA?pid=ImgDet&rs=1)

[9] Felius, M.; Koolmees, P.A.; Theunissen, B.; Lenstra, J.A. On the Breeds of Cattle—Historic and Current Classifications. Diversity 2011, 3, 660-692.

[10] Li G, Chen Z, Purswell J, Linhoss J. Practices and applications of convolutional neural network-based computer vision Systems in Animal Farming: a review. Sensors. 2021;(21:(February).

[11] Training, testing, validation image source (https://www.v7labs.com/blog/train-validation-test-set).

[12] Max & Avg Pooling(https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer).

[13] Kaur S. Singh, A.K. Singh, Computer Vision-Based Approach for Automatic Detection of Dairy Cow Breed.

[14] Qiao Y, Su D, Clark C, Lomax S, Clark C. Individual cattle identification using a Deep Learning Based Framework. IFAC PapersOnLine. 2019;52(30):318-323. doi:10.1016/j.ifacol.2019.12.558.

[15] Jwade SA, Guzzomi A, Mian A. On farm automatic sheep breed classi fi cation using deep learning. Comput Electron Agric. 2019;167(October):105055. doi:10.1016/j.compag.2019.105055.

[16] Dutta R, Smith D, Rawnsley R, et al. Dynamic cattle behavioural classification using supervised ensemble classifiers. Comput Electron Agric. 2015;111:18-28. doi:10.1016/j.compag.2014.12.002.

[17] Benos L, Tagarakis AC, Dolias G, Berruto R, Kateris D, Bochtis D. Machine learning in agriculture: a comprehensive updated review. Sensors. 2021;21(4):1-55. doi:10.3390/s21113758.

[18] Aksoy, S.; Yilmazturk, F.; Yilmazturk, F. Detection and Breed Classification of Cattle Using YOLO v4 Algorithm. In Proceedings of the 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 25–27 August 2021; pp. 1-6.

[19] Contemporary Machine Learning Applications in Agriculture: Quo Vadis? Atif Mahmood, Amod Kumar Tiwari, Sanjay Kumar Singh, Sandeep S. Udmale. DOI: 10.1002/cpe.6940.

[20] Dropout applied to a Standard Neural Network (Image by Nitish) https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9.

[21] Feed Forward Neural Network with input layer, output layer and 2 hidden layers (https://www.researchgate.net/figure/Feed-Forward-Neural-Network-with-input-layer-output-layer-and-2-hidden-layers_fig1_332545886).

[22] Bahbahani H., Tijjani A., Mukasa C., et al.Genetic structure and phylogeography of African indigenous zebu cattle populations revealed by patterns of variation in mitochondrial DNA sequences.Genetics Selection Evolution2017;49(1):85.