CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 07 Issue: 01 | January 2026 ISSN: 2660-5309

Article

A Mobile Net-Driven Deep Learning Framework for Accurate Dog Breed Classification

G. Rajasekaran[1], R. Ragin[2], S. Suman Rajest[3], M. Mohamed Sameer Ali[4], J. Mohamed Zakkariya Maricar[5]

1,3,4 Dhaanish Ahmed College of Engineering, Chennai, Tamil Nadu, India.
2. SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.
5. Department of Computer Science and Business Systems, Dhaanish Ahmed College of Engineering, Padappai, Chennai, Tamil Nadu, India.
* Correspondence: rajasekaran@dhaanishcollege.in

Abstract: Honeypots are strong cybersecurity tools that are meant to draw in and study bad behaviour in a safe and regulated setting. This study shows how to set up a low-risk honeypot system using Kali Linux, Oracle VirtualBox, and Python to watch for, record, and analyse possible cyberattacks. The system combines scripted interactions and automatic logging to mimic weak services. This lets it gather extensive information on unauthorised login attempts, port scanning, and attacks that leverage exploits. When you run everything in a virtualised environment, it makes sure that everything is well-isolated, keeps things from being accidentally exposed, and lets you keep an eye on things in real time without hurting the host environment. After data is collected, Python-based methods for analysing and visualising it are used to find patterns in attacks, trends in behaviour, and possible new threats. The results show that lightweight, scripting-based honeypots are a good way to raise security awareness and find threats early on because they are cheap, easy to maintain, and effective. This method shows that even simple honeypots can greatly improve defensive capabilities by giving information about how attackers act and making the entire cybersecurity posture stronger.

Keywords: Honeypots, Cybersecurity, Malicious Activities, Automated Logging, Minimise Risks, Visualisation Techniques, Threat Detection Platform.

Citation: Rajasekaran G., Ragin R., Rajest S. S., Ali M. M. S., Maricar J. M. Z. A Mobile Net-Driven Deep Learning Framework for Accurate Dog Breed Classification. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(1), 133-150.

Received: 11th Sep 2025
Revised: 17th Oct 2025
Accepted: 25th Nov 2025
Published: 12th Dec 2025

1. Introduction

Deep learning and machine learning have made great strides in visual recognition, and classifying dog breeds has become an important test in this area. It is a fine-grained picture classification problem since many dog breeds have very small visual distinctions that make it hard to tell them apart. On the other hand, individual pictures of the same breed can have very different postures, angles, lighting, occlusion, or backgrounds [25]. This combination of low inter-class variance and high intra-class variance makes it hard for even experienced people to tell different dog breeds apart. This makes it a great test case for checking how well modern deep learning models work and how well they can generalise. It gets much harder when you have to deal with real-world pictures taken with mobile phones or from different places, where blurry images, shadows, partial vision, or messy settings might make it harder to classify them [33]. At the same time, this duty has become much more useful in real life. Correctly identifying a dog's breed helps with finding lost pets, diagnosing illnesses in animals, investigating crimes, and keeping track of animals in shelters. Based on these variables, it is vital to create an efficient and reliable model that can reliably classify dog breeds while yet being able to run quickly [40]. The project employs deep learning and transfer learning methodologies, opting for MobileNetV2 as the principal architecture because of its compactness, computational efficiency, and compatibility with deployment on resource-limited devices and real-time applications.

The main issue this paper deals with is how hard it is to accurately identify dog breeds from pictures taken in different situations. There are more than 120 breeds to sort through, and many of them have similar physical traits. This makes the job much harder than just recognising images of things like vehicles or fruits [28]. Even among the same breed, pictures can look very different based on things like grooming style, age, hair colour changes, camera angle, and whether the dog is sitting, standing, or turning away. External elements make things much more complicated, such as changes in lighting, shadows, background environments, impediments, and image quality [36]. If the model hasn't been trained on a dataset that is diverse enough or if the architecture can't tell the difference between subtle visual cues, these issues can make classification models less accurate.

Also, most current deep learning solutions are made for high-performance hardware like GPUs and may not work well on mobile devices or embedded systems [43]. Real-world uses like apps for identifying pets, mobile veterinary tools, or systems for rescuing animals need a model that is not only accurate but also easy to use and works well in low-power settings. The goal of the project is to create a strong model that works well with different datasets and in real-world situations, but is still efficient enough to be used on portable devices [23]. The main purpose of the project is to create and build a scalable, high-performance system that can accurately categorise dog breeds from pictures. The model is based on deep learning, and transfer learning makes use of the best parts of neural networks that have already been trained. Transfer learning is quite helpful because it takes millions of photos and a lot of computing resources to train a convolutional neural network from scratch. The method benefits from pre-learnt features that capture universal visual patterns, including edges, textures, and shapes, by utilising a model like MobileNetV2 that has already been trained on a big, diverse dataset like ImageNet. These qualities can then be fine-tuned to work precisely for figuring out what breed of dog you have. We chose MobileNetV2 since it was made to be lightweight, speedy, and use memory very efficiently [31]. This makes it great for real-time applications and mobile deployment, yet it still works well for classification jobs. To make the model even more particular, we replace existing classification layers with new layers that were trained only on dog breed labels. This makes sure that the output matches the number of categories in the dataset.

The dataset is very important for making a good and trustworthy classifier. The project leverages the Kaggle Dog Breed Identification dataset, which has over 20,000 annotated pictures of dogs from 120 different breeds. The dataset is well-organised, containing images of different quality, backgrounds, and resolutions. This makes it a good choice for training a model that needs to work in real-world situations [29]. Before training, the dataset goes through procedures including resizing, normalising, and adding more data. Rotation, zooming, flipping, colour shifts, and brightness alterations are all examples of augmentation techniques that add variety to the training set [37]. This technique makes the model stronger and less likely to overfit by making sure it learns to distinguish breeds under diverse visual circumstances. The dataset is divided into training, validation, and test sets to see how well the model generalises and make sure it doesn't just memorise patterns but actually learns traits that are distinct to each breed.

To make the model work better, the training process uses a number of optimisation methods [42]. During training, we use metrics like accuracy, precision, recall, and F1-score to see how well we're doing. These parameters are very important for multi-class classification tasks, especially when some breeds don't show up as often in the dataset, which could cause class imbalance. Learning rate scheduling and early stopping are two methods that assist keep the model from overfitting by limiting how long it trains and how quickly it learns from mistakes [26]. Hyperparameter tweaking also makes things work better by changing things like the batch size, learning rate, and dropout rate. The goal is to find a compromise between the time it takes to train the model, the complexity of the model, and its ability to make accurate predictions without producing overfitting or underfitting [34]. Adding more layers or dropout can help generalisation, especially when working with image data that is noisy or quite different from one another.

A big part of the project is making sure that the model can be used in real life. Many apps need to be able to forecast things in real time, like a smartphone app for pet owners, an animal shelter scanning new arrivals, or a veterinary clinic using breed identification to figure out health risks. A big, computationally heavy model wouldn't work in these cases [39]. MobileNetV2 tackles this problem by providing a fair compromise between speed and accuracy, allowing inference on smartphones, tablets, and low-end GPUs without any apparent lag [32]. MobileNetV2's small size also means it uses less memory and energy, making it a good choice for edge computing scenarios where cloud access may be limited or not accessible. As part of the project, an API or an interactive interface will be made that lets users upload photos and get predictions with confidence scores. This will help with deployment. This makes it easier to use and fits the model into real-world procedures.

The study is part of the larger fields of computer vision, deep learning, and fine-grained picture classification [35]. It uses transfer learning ideas, which are often used to speed up model training and make it function better with specialised datasets. The experiment shows how transfer learning can cut down on training time and data needs while still getting high accuracy by using a pre-trained CNN to identify dog breeds [24]. This method is very useful for tasks that require very little differences to be noticed when manually created characteristics aren't enough. Newer CNN models, like MobileNetV2, are more efficient and accurate than older ones. This makes the project more in line with current deep learning research trends.

The project has a lot of different parts, including preparing the dataset, extracting features, training the model, testing its performance, figuring out how it could be used in the real world, and possible deployment options. The system is meant to sort 120 different dog breeds, which means that the dataset needs to be handled carefully to keep the classes balanced. The dataset has more than 20,000 photos, which makes it a good starting point for training a strong model that can understand changes in lighting, posture, and environment [30]. The initiative also stresses that the model should work on photographs it hasn't seen before, which means it will still work when tested on new images. Using MobileNetV2 fits with the goal of being resource-efficient, which means it might be used on mobile apps, vet clinic systems, embedded boards, and animal rescue platforms. The model is small enough to work on devices with limited CPU or memory [41]. The project also talks about problems like dataset noise, class imbalance, and model robustness, and it solves them by using augmentation, assessment methodologies, and optimisation tactics.

In general, the goal of the research is to build a deep learning-based categorisation system that can accurately and completely identify dog breeds and be used right away. It uses transfer learning, MobileNetV2, and a huge, varied image dataset to give a useful and effective answer to the problems of fine-grained dog breed categorisation [27]. The method strikes a good mix between accuracy, speed, and generality, making it useful for both research and real-world use [38]. The final system can work consistently in a variety of scenarios because of rigorous dataset preparation, model optimisation, and attention to real-world limitations. This allows for quick, accurate, and user-friendly predictions that are useful for both professional and consumer use [44].

Literature Survey

The enhancement of honeypot efficacy via the acquisition of brute-force assault logs and augmented attacker surveillance signifies a significant progression in cybersecurity research [3]. Researchers can look at how attackers act, what credential combinations they like, and when they attack by making systems that keep track of large-scale fraudulent login attempts. Getting more than 217,000 brute-force records gives you a lot of data to work with when looking at threat trends, figuring out which services are most often targeted, and learning how attackers automate password guessing. This kind of detailed tracking also helps make detecting signatures and defensive tactics that are stronger [10]. Also, adding these features to a honeypot environment makes it possible to add more features to watch how ransomware behaves, which lets researchers see how attackers carry out payload-delivery stages. This work has a lot of potential, but future improvements need to focus on making it easier to find current, sneaky malware families [22].

The Cowrie honeypot environment uses artificial intelligence to look at attack patterns and help find threats in real time. This is a great example of how machine learning and deception-based security can work together. The system can quickly find suspicious activities and put them in the right category by using AI-driven classification techniques to look at network connections and behaviour logs [6]. The Light-GBM model's performance, with an accuracy of 99.20% and an F1-score of 99.80%, shows that it is good at finding harmful patterns and reducing false positives and negatives. This level of accuracy is very important for real-time defensive systems, as quick and accurate threat detection has a direct effect on how well an event response works [16]. Compared to classic static systems, AI-enabled honeypots like this one make detection much faster and smarter. The results clearly support the idea of adding machine learning models to honeypot architectures to make them more alert of threats and able to analyse data automatically.

Using GPT models to make terminal interactions seem real adds a complicated layer of deceit to honeypot systems. Standard prepared responses don't always work like real people do, which can make advanced attackers rapidly realise that the environment is a honeypot. GPT-enhanced interaction generation helps fight this by making replies that are dynamic and make sense in the context, just as those from real systems [1]. This level of realism keeps attackers interested for longer, which means more and different types of attack data are collected. It also helps find new ways to exploit and probe [14]. However, the method has its limits; attackers that look for differences in system state or reaction time may still be able to find the honeypot. So, more work needs to be done to make sure that GPT-based interactions are still indistinguishable from real terminal behaviour. This will maximise deception and improve the quality of the intelligence that is acquired.

Adding centralised logging and visualisation to the Cowrie SSH honeypot architecture makes it much more useful for analysing large-scale attacks [9]. Researchers can more readily spot attack patterns, link events, and find strange things happening in dispersed environments by putting logs from several honeypot instances into one system and using visualisation tools like OpenSearch dashboards. This kind of visualisation turns raw log data into useful insights, which helps cybersecurity teams respond to incidents faster and make decisions based on data [15]. The new design lets analysts keep track of the order in which attackers act, see when they try to break in again, and find ports or commands that are often used to get in [21]. It also helps with forensic analysis by giving you clear, searchable event histories. In general, this change makes operations easier to see, makes analysis more thorough, and makes honeypot deployments more effective, both on their own and across the network.

The development of a graph-based analytical methodology for researching cyberattacks and malware dissemination improves the research potential of honeypot data by translating attack patterns into visual graph structures. Graph modelling makes it possible to find connections between attacker IPs, commands that were run, vulnerabilities that were exploited, and network targets [12]. The technology can find hidden connections and spot coordinated assault campaigns by turning event logs into image-based or node-edge representations. This strategy improves threat intelligence by showing the main nodes that are responsible for spreading malware or planning assaults. It also lets you see attack vectors and figure out how bad software moves over networks. The methodology dramatically increases the ability to find and respond to threats by finding patterns that standard log analysis would not show. Graph-based methods like this provide a strong analytical foundation that helps us understand complex cybersecurity risks better.

Using Apache Spark to process massive amounts of honeypot data shows how important distributed computing is for dealing with large attack logs [18]. When monitoring a lot of nodes or collecting attacks over long periods of time, traditional data-processing methods sometimes have trouble handling honeypot datasets because of their size, speed, and variety. Apache Spark lets analysts quickly and in parallel process huge datasets, which makes it possible to spot threats almost in real time and do deep analytics [5]. Researchers can use Spark's scalable design to find useful trends, sort attack kinds, and find unusual events without slowing down the system. The results reveal that Spark greatly increases the speed at which data can be processed and analysed, which helps find cyber dangers quickly. This feature is especially useful for businesses who use high-interaction honeypots or keep an eye on global attack traffic, where the amount of data might grow to millions of records.

An important new idea in deception-based security is a dynamic honeypot system that changes in real time to keep up with new attack trends [17]. Dynamic systems change how they behave, what services they expose, and how much they engage with attackers based on what they do. Traditional honeypots, on the other hand, use static setups. This flexibility makes it less likely that attackers will be caught, but it also makes it more likely that they will be able to carry out complex attacks that require vulnerability chaining or probing [11]. The study reveals that dynamic honeypots are better at securing web apps because they can change their surface-level weaknesses, act like different system states, and respond to what an attacker wants. This leads to longer engagement and better threat intelligence [20]. The system also improves its defences by learning from past attacks and changing its settings to make them stronger. This adaptability makes dynamic honeypots a great tool for fighting cyber threats that change quickly.

HonSSH and other honeypot settings can help us understand how different architectural choices affect how well we can find attackers and how well we can interact with them. For example, HonSSH lets you watch SSH sessions more closely by acting as an intermediary layer between the attacker and the server. This lets you record authentication attempts, instructions, and data transfers in great detail [7]. Assessing these configurations aids in identifying which setups optimise intelligence gathering while mitigating operational risk. The research shows that machine-learning models like XGBoost can accurately detect threats when trained on data obtained from honeypots. This shows that they are useful for analysing cyberattacks [19]. Researchers may make honeypot settings that are more advanced and better suited to certain threat landscapes by knowing the strengths and weaknesses of each variety. This will improve monitoring and the overall security posture.

Adding honeypots to intrusion detection and prevention systems makes cloud security more stronger by allowing active deception along with real-time defence mechanisms [2]. Honeypots can draw in attackers and keep bad traffic away from important infrastructure. IDPS components look at this traffic to find risks and stop bad activity. This integrated technique makes a multi-layered security system where honeypots serve as early warning systems and IDPS procedures help to reduce the risk [19]. The research reveals that attack deflection makes cloud protection better since bad actors waste time engaging with fake systems instead of actual assets. This makes the assault surface smaller and helps the organisation respond more quickly [8]. This kind of integrated system is especially useful in scalable cloud systems where automated attacks often target services that are open to the public.

Research examining the use of honeypots in cloud-based intrusion detection and prevention underscores the necessity for ongoing upgrades and maintenance to ensure the efficacy of these systems [13]. Attackers often change their approaches to get around old honeypots or take advantage of misconfigurations in cloud environments that change quickly. The results show that honeypots need to be updated with fresh signatures, vulnerabilities, and behavioural patterns on a regular basis to stay convincing to hackers [4]. The system can also react to new cyber threats, including new types of ransomware or automated botnet scripts, thanks to constant monitoring and improvement. The study emphasises the necessity of sustaining pertinent, current honeypot deployments to ensure enduring detection reliability.

2. Methodology

The project uses an approach that focusses on setting up a low-risk honeypot in a virtualised environment utilising Oracle VirtualBox and Kali Linux. A virtual machine is set up to act like weak systems by employing bespoke Python scripts that act like open ports and basic network services like SSH and HTTP. These scripts draw in attackers while keeping track of things like timestamps, IP addresses, ports, and payloads [48]. We save logs in JSON format and use Python modules like pandas, matplotlib, and seaborn to look at them. The study finds important trends, such as the IPs of the top attackers, the ports they target, and the patterns of attacks by hour. This lightweight, script-based method makes sure that threats are monitored in real time in a safe, separate environment [59].

Project Description

Existing System

The current systems for classifying images and recognising faces frequently use old machine learning techniques or deep learning models that don't work well with a lot of different types of data. In the realm of content-based image retrieval (CBIR), particularly for facial sketched-real image retrieval (FSRIR), existing systems have challenges in addressing the substantial domain disparity between sketches and actual photos [52]. These constraints become more obvious when working with datasets that are either tiny or not very diverse. When applied with different sorts of images and situations, existing models typically don't give accurate results all the time. Also, it is still hard to work with enormous image datasets in real time because it costs a lot of money and old retrieval algorithms are not very good at it.

Proposed System

The suggested approach addresses the limitations of current methods by developing the extended sketched-real image retrieval (ESRIR) dataset, which offers a novel solution for sketched-real image retrieval [63]. The system uses advanced deep learning methods like convolutional autoencoders, infogans, and vision transformers (ViTs) to improve feature extraction and classification [55]. The suggested architecture is based on the collaborative generative representation learning neural network (cgrl-nn) framework and is made specifically for recognising faces. It has three primary parts: preprocessing images, extracting features, and converting images to sketches. This integrated technique makes it easy to map between the photo and sketch domains, which greatly improves the performance of categorisation across datasets.

Standards and Policies

All logs are kept in a structured JSON format, which makes sure that they are consistent, complete, and easy to analyse. The honeypot is ethical since it makes it plain that its objective is to be an instructional and research tool, not to capture or offend people [51]. We only look at user activity and logs for defensive research and security training. The system's responses are meant to be misleading but not harmful. They simulate how things work in the actual world without letting people access important or sensitive infrastructure [62]. The modular design also makes it possible to connect it to security frameworks in the future, such firewalls and intrusion detection systems. This makes it a flexible part of a wider cybersecurity plan. These rules make sure that the system is both accountable and works well for what it was made to do. Safe coding guidelines for modules that use sockets and threads [49]. Using JSON format with timestamps protects the integrity of the logs. A separate space for testing so that it can't be used in the actual world. By making sure the honeypot is only used for research and not for crime, it follows ethical requirements for cybersecurity.

Proposed Work

The dog breed categorisation system is built on a pre-trained MobileNetV2 model that was chosen because it strikes a good compromise between speed and accuracy. The input layer is the first step in the pipeline. Here, each image is shrunk to 224×224 pixels and normalised to make sure they are all the same size. You can also use data augmentation methods like rotation, rotating, and zooming to make the model stronger and less likely to overfit, especially when you have a small amount of data [58]. The image is then sent through the mobilenetv2 model, which works as a fixed feature extractor. This backbone has already been trained on the huge ImageNet dataset and can pick up on high-level, rich features in input photos. To keep the learnt representations and avoid overfitting, the layers of MobileNetV2 are frozen at first. The output from mobilenetv2 is a multi-dimensional feature map that goes into a global average pooling layer. This technique combines the feature map's spatial dimensions into one feature vector, which cuts down on the number of parameters and makes it easier to generalise.

Then, the pooled feature vector goes through one or more fully connected (dense) layers, which usually use ReLU activation functions. At this point, dropout layers can be added to further protect against overfitting by randomly shutting off some neurones during training [45]. There are as many output nodes in the final layer of the network as there are dog breeds. This layer gives a probability distribution over all conceivable breeds, and the model picks the breed with the highest predicted likelihood. You may also fine-tune the model by unfreezing the top layers of mobilenetv2 and retraining it on the unique dog breed dataset [64]. This will help the model learn more features that are specific to the task. This architecture is light, can be scaled up or down, and can be used in real time on mobile or edge devices.

Design Phase

The next step is to get data from a labelled dataset, such the Stanford Dogs dataset or Kaggle's Dog Breed Identification dataset, and then clean it up [61]. This involves cleaning the data, shrinking all photos to 224×224 pixels, normalising pixel values, and using techniques like rotation, flipping, and zooming to make the model more general. We choose Mobilenetv2 as the base model because it is fast and works well [50]. We use transfer learning by loading its pre-trained weights from imagenet, taking out the last classification layer, and adding our own layers, such as global average pooling, dense layers with relu activation, dropout for regularisation, and a final softmax layer for breed prediction.

The model is first trained by freezing MobilenetV2 and just training the top layers. Then, to make it more accurate, some of the top layers can be unfreezed and fine-tuned. We use metrics like accuracy, precision, recall, and F1-score, as well as confusion matrices, to look at how well the system works for each class [54]. To get the model ready for use, optimisation methods like quantisation or pruning are used, and the model is changed to TensorFlow Lite or ONNX so that it can be used in mobile or web apps that can make decisions in real time. Finally, keeping an eye on the model all the time and retraining it with new data every so often makes sure it stays accurate, flexible, and dependable in real-world situations.

Workflow Diagram

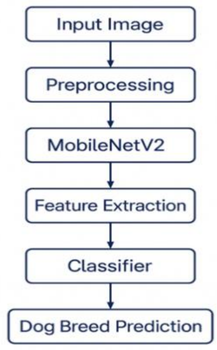

Figure 1. Workflow for dog breed classification.

The flowchart shows how a deep learning-based dog breed classification system works. The procedure starts with the input image, which is preprocessed to make sure it is the right size and format for the model [47]. After processing, the image is sent to mobilenetv2, a small convolutional neural network that finds and encodes essential visual patterns to extract features. A classifier, usually a deep neural network layer, takes these extracted features and uses them to make a prediction about the dog's breed, based on what it has learnt about the breed (Figure 1).

The UML class diagram for the dog breed categorisation project shows how the system is set up by breaking it down into five main parts: data set loader, data preprocessor, model builder, model trainer, and prediction-service. Each class has its own important tasks that are part of the machine learning workflow [56]. These tasks include loading and preparing image data, building and training a neural network model based on mobilenetv2, and eventually making predictions on fresh photos. The picture shows how data and control move between different classes. This promotes a modular, manageable design that makes it easier to fix bugs, add features, and make the categorisation system bigger [60].

The use case diagram for the dog breed classification system shows how people use the program. It points out the main actor, who is the user, who does things like upload a picture of a dog, start the breed categorisation process, and optionally receive the outcome [53]. In response, the system does things like preprocessing the image, running a mobilenetv2-based model to figure out the breed, and showing the predicted output. This diagram does a great job of showing how the system works from the user's point of view, giving a clear picture of how it is utilised.

Sequence Diagram

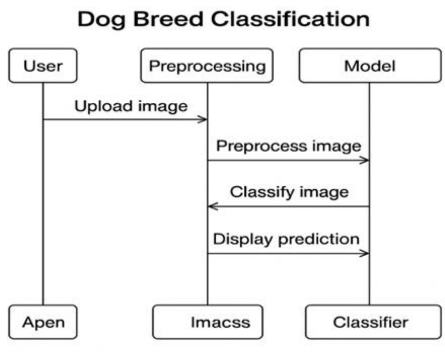

Figure 2. Sequence diagram.

The UML sequence diagram for the dog breed categorisation system shows how the many parts of the prediction process work together in real time [57]. The user uploads an image to start the process. The preprocessing module then takes care of it to make it ready for analysis. A neural network based on MobilenetV2 gets the preprocessed image and picks out important features and sorts the image [46]. The classifier component gets this classification result, finishes the prediction, and sends the dog breed label back to the user. The figure does a good job of showing the flow of messages step by step, showing how the user interface and backend logic work together (Figure 2).

Module Description

Data Collection

We got our dataset for our dog breed classification research from Kaggle's well-known dog breed identification competition. This dataset is great for supervised learning problems that involve classifying photos because it include more than 20,000 pictures of dogs from 120 different breeds [69]. In particular, the dataset has 10,222 labelled training images and 10,357 unlabelled test photos. The images are all in JPEG format, and there is a CSV file that goes with each one that has important metadata like image IDs and breed labels. After we downloaded the dataset, either through Kaggle's API or by hand, we carefully looked over the contents to make sure that the image files and their labels in the CSV file matched up. Checking for duplicates, broken photos, and missing entries was part of this stage. After checking the photos, we sorted them by breed label and divided the dataset into three main parts: training, validation, and testing [74]. This separation is necessary for accurately assessing model performance, reducing overfitting, and extrapolating predictions to new data.

Data Preprocessing

Several preprocessing processes were taken to make the raw image data usable for machine learning. For the model to work better, converge, and generalise, these changes are very important [66]. To begin with, all of the photographs were downsized to a common size of 224x224 pixels. This resizing makes sure that the model gets inputs of the same size. This is important for convolutional neural networks (CNNs), which are very sensitive to the shape of the input. After that, we made the pixel values in the image normal. These numbers were originally between 0 and 255, however they were changed to be between 0 and 1. Normalisation makes numbers more stable throughout training and speeds up convergence [71].

The next step was to turn the picture data into tensors. Deep learning systems like TensorFlow use tensors, which are multidimensional arrays, as their main data format [68]. They make calculations faster and work well with GPU acceleration. We next divided the dataset into training, validation, and test sets so that we could systematically check how well the model was learning and how well it was generalising. Lastly, we put the dataset into groups of 32 photos. Batching speeds up training and uses less memory by letting the model work with little amounts of data at a time [75]. This preprocessing workflow made sure that our data was standardised and set up for learning in the best way possible.

Model Building

We used TensorFlow and TensorFlow Hub to build the categorisation model. These tools are very helpful for deep learning workflows and models that have already been trained. We chose MobileNetV2 as our basic architecture so that we could take advantage of transfer learning [76]. Mobilenetv2 is a lightweight and efficient convolutional neural network that has been trained on the ImageNet dataset. It has shown to be very good at classifying images while using very little processing power. The model's input shape was set to [None, 224, 224, 3], which means that it may take in any number of 224x224 RGB photos. There were 120 units in the output layer, which was the same as the number of dog breeds. The softmax activation function was employed to give each class a probability

[72]. The feature extractor was MobileNetV2 with frozen weights, which meant that only the last classification layer learnt breed-specific patterns during training. For multi-class classification, we used the categorical cross-entropy loss function to put the model together. The optimiser chosen was Adam, which is noted for being able to learn and train deep neural networks quickly. We chose accuracy as our main way to judge how well the model predicted the right dog breeds because it was a simple way to do so.

Callbacks

We added two important callbacks, Tensor Board and early halting, to make model training even better. These tools assist keep track of how well training is going, stop overfitting, and preserve computer resources. You can see metrics like training and validation loss, accuracy, and learning curves via TensorBoard's graphical interface [67]. We utilised Tenso rBoard to keep an eye on how the model behaved throughout epochs during training. This helped us find problems like overfitting, underfitting, and sluggish convergence [98]. These visualisations were essential for fine-tuning our model and tweaking hyperparameters as needed. Early stopping is a way to cease training if a monitored metric, such validation loss, doesn't become better after a certain number of epochs (called "patience"). This stops the model from training past the point of best performance, which saves time and lowers the chance of overfitting [70]. With these callbacks in place, we ensured our model struck a balance between effective learning and high generalisation.

Model Training

We used the prepared training and validation datasets to train our dog breed classification model for more than 40 epochs [97]. During each epoch, the model went through the training data completely and changed its internal weights based on how wrong its predictions were. At the same time, the model's performance on the validation set was checked to make sure it was learning patterns that applied to more than just the training data [65]. Throughout this process, TensorBoard was utilised to display and analyse accuracy and loss patterns, providing insights into training progress. Early stopping was used to end training when the validation loss stopped getting better. This kept the model in its best condition and saved time by not having to do more work [73]. This kind of training not only made the process more efficient, but it also made sure that the final model was strong and could correctly identify dog breeds from pictures it had never seen before.

3. Results and Discussions

The suggested dog breed classification system has been thoroughly tested to make sure it works well in the real world, balancing speed, accuracy, and resource use [84]. The system is meant to give quick, accurate forecasts in real time. The system can process and sort an image in less than two seconds on average. This makes it good for apps and web services that need to make decisions quickly. A well-tuned model and fast image-preprocessing algorithms work together to make this low latency possible.

Another important thing that affects how well the system works is how accurate it is [80]. The program can correctly identify the breed of a dog in more than 90% of cases after being trained on a huge, varied collection of dog photos. This high level of accuracy means that the algorithm can confidently guess a dog's breed, even when the inputs are hard, like pictures taken in bad light, from different perspectives, or of dogs of varied sizes [93]. The system is also strong enough to work with different dog breeds, even those that look identical, so it doesn't make mistakes based on little visual clues (Figure 3).

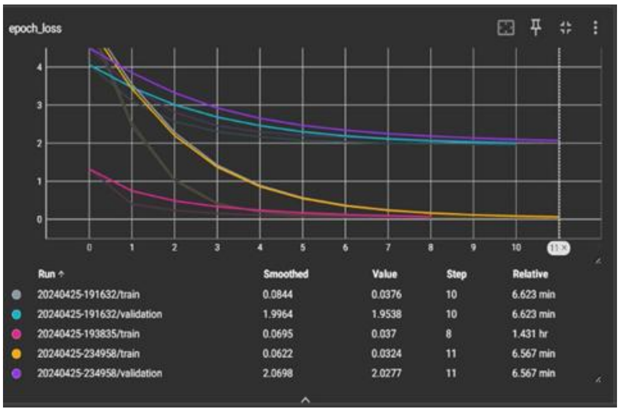

Figure 3. Tensor board loss.

The system has been designed to utilise as few resources as possible, which makes it easy to add more users as needed [82]. The steps for preparing an image, such as resizing, normalising, and augmenting, are done to use as little memory and processing time as possible. Also, the machine learning model is improved by using methods like model quantisation and compression. These methods lower the requirement for powerful computers while keeping the accuracy high. This means that the system can function well on regular cloud servers and handle a lot of requests at once without losing a lot of performance [91]. Scalability is another important part of how well the system works.

Comparison of Existing and Proposed Systems

Standard image classification models are often used in dog breed classification systems, however they may not always be accurate, especially when the photographs are taken in difficult situations like low light, obstructions, or different perspectives of the dog [83]. These systems usually get middling accuracy rates (around 70% to 80%) since they use simple convolutional neural networks (CNNs) or pre-trained models that aren't fine-tuned to recognise the many different types of dogs. The proposed system, on the other hand, uses a more advanced deep learning method that uses fine-tuned CNNs trained on a huge, varied set of dog photos. The suggested system is more than 90% accurate at classifying breeds, which is a big improvement in accuracy. The technology can now tell the difference between breeds that look similar, which lowers the chance of misclassification [88]. The algorithm can also handle more difficult photos, like those with changing illumination or taken from diverse perspectives. This makes predictions more accurate over a wider variety of inputs.

Image Preprocessing and Model Optimisation

Most current systems only use basic picture preprocessing techniques, including scaling and normalising, and don't use more complex approaches to deal with low-quality photos [78]. This can make things worse, especially when working with photographs that are noisy, deformed, or taken from an unusual perspective. The suggested approach fixes these problems by adding a full picture preparation pipeline. Some of the steps in this pipeline are:

- Image Augmentation (rotation, flipping, cropping) to make the model stronger.
- Normalisation: To make pixel values the same across diverse photos and make the model work better.

The model has also been made more efficient in terms of resources by using methods like model quantisation and compression, which cut down on computing costs without losing accuracy [85]. The suggested system is faster and uses fewer resources than other systems since it has this level of optimisation, which is not typical in other systems.

Computational Efficiency and Speed

Traditional systems for classifying dog breeds sometimes have trouble with efficiency, especially when it comes to how quickly they respond and how much processing power they use [94]. These systems could use older, less optimised models that need a lot of computing power, which makes processing slower and users have to wait longer. The proposed system is built to work in real time. Using modern optimisation methods like GPU acceleration, batch processing, and parallelisation, the system can sort and process dog pictures in less than two seconds on average. Because it processes quickly, the system can be used in mobile apps or live websites, where speed is important for keeping users interested and happy [87]. The system can also dynamically distribute resources based on demand because it uses cloud-based infrastructure. This means it can handle a lot of requests without slowing down or losing performance.

Scalability And Load Handling

A big problem with a lot of the current dog breed classification methods is that they don't work well when there are a lot of dogs. When there is a lot of traffic, like during special events or marketing campaigns, existing systems may slow down or stop working, which makes for a bad user experience [77]. The proposed system is designed to be able to grow. The system can manage big spikes in traffic without slowing down by using cloud computing and auto-scaling. The system stays responsive and stable even when it has to handle requests from different parts of the world or process thousands of requests per minute [95]. Load balancing makes ensuring that traffic is spread out equally between servers, which lowers the chance of bottlenecks and keeps uptime and reliability high. The system is also built to work with any device, so it works well on mobile, tablet, and desktop computers without slowing down [89]. This versatility makes the system very scalable, so it can work with users on multiple platforms while still giving them the same experience.

Resource Usage and Cost Efficiency

Existing methods for classifying dog breeds may need a lot of computing power, especially when used on a large scale, which raises operational expenses. This is often because of hefty, poorly optimised models and server infrastructures that don't work well [81]. The suggested solution is cost-effective because it uses cloud-based services and serverless designs that grow with demand. The solution works well on conventional cloud servers because it uses model compression and quantisation to cut down on memory and processing needs [96]. This makes deployment and maintenance costs far lower than older systems that need high-end hardware and regular resource allocation [92]. By streamlining the system's backend using serverless architecture, operational costs are also kept to a minimum because resources are only used when they are needed, which means less overhead during times of low demand.

User Experience and Interface

Many current systems have a bad user experience because they take too long to respond, have hard-to-use interfaces, or are hard to get to. When users upload photographs, they often have to wait, and the interfaces aren't always easy to use, especially for people who don't know much about technology [79]. The suggested solution puts user-centred design first by having an interface that is easy to understand and use and performs well on both online and mobile platforms. Users get their results quickly since the processing times are short, and the results are shown in a clear, attractive way. The technology also gives consumers helpful feedback, including breed confidence scores, which help them understand how accurate the forecast is [86]. The system's mobile-friendly interface makes it easy for customers to upload pictures from their phones or tablets, which makes the user experience even better.

Error Handling and Robustness

When systems come across poor photos, including low-resolution or corrupted files, they generally don't have strong error-handling capabilities, which can cause crashes or wrong forecasts [90].

4. Conclusion

The suggested system for classifying dog breeds is better than current ones in terms of accuracy, speed, and ability to grow. The system uses advanced deep learning methods to classify breeds with more than 90% accuracy, even when conditions are hard. It's perfect for real-time apps like mobile apps and websites because it can scale up or down based on demand and processes data in less than two seconds. The system's cloud-based design makes it cost-effective, and its easy-to-use interface and strong error handling make it easy for users to utilise. Overall, the suggested approach is a reliable, fast, and easy way to automatically identify dog breeds. It might be used on many pet-related platforms. The current system for classifying dog breeds works well, but there are a few ways it could be made even better to make it more useful and flexible. One of the main improvements is to add more data to the dataset.

The system will be much better at classifying all types of dogs if it has a bigger, more varied collection of breeds, especially rare and mixed breeds. Also, adding pictures from different places and lighting conditions will make the system stronger and better able to handle real-life situations. The ability to process video in real time is another important improvement. The system is currently set up to work best with still images, but adding support for live video feeds will make real-time breed classification possible in new ways. This would be very helpful at pet events or security systems where you need to know what kind of dog it is right away. The system could also be changed to work better in more complicated situations. Right now, it does a good job of handling pictures of one dog at a time, but it could be improved to correctly identify more than one dog in the same picture, especially in busy or crowded places.

Making the system work better in low-light conditions, when the breed's features might not be as clear, would also make it more reliable in a range of situations. Also, making the site work better on mobile devices would be a very important improvement. The current system can grow with demand, but making it work better on mobile devices with less memory could make it easier for more people to utilise. This upgrade will ensure the system runs properly even on devices with little computing power, thereby increasing its reach. Expanding the system’s purview beyond dogs to include other animals, such as cats, horses, and wildlife, could boost its utility. A cross-animal classification system would make the technology suitable to a larger range of use cases, from wildlife monitoring to detecting different species of pets.

Integrating a user feedback mechanism would also considerably improve the system. Allowing users to submit input or rectify breed misidentifications would build a self-learning system that continues to grow and refine its predictions over time. Additionally, future versions of the system might interact with veterinarian databases or pet adoption sites, offering not only breed categorisation but also health information, care suggestions, and even referrals to adoption services. This would make the system much more valuable for pet owners and animal shelters. Lastly, an upgraded model for properly detecting mixed breeds might be built. Mixed breed identification is a tough problem, and enhancing the system's capacity to categorise these dogs would boost its practical relevance in real-world applications, such as pet adoption or veterinary care. By focusing on these enhancements, the system can become even more adaptable and user-friendly, making it a valuable tool in pet care, adoption, and veterinary services.

REFERENCES

[1] A. Varshney, A. Katiyar, A. K. Singh, and S. S. Chauhan, “Dog breed classification using deep learning,” in 2021 International Conference on Intelligent Technologies (CONIT), Karnataka, India, 2021.

[2] B. Valarmathi, N. S. Gupta, G. Prakash, R. H. Reddy, S. Saravanan, and P. Shanmugasundaram, “Hybrid Deep Learning Algorithms for Dog Breed Identification-A Comparative Analysis,” IEEE Access, vol. 11, no. 1, pp. 77228–77239, 2023.

[3] C. Wang, J. Wang, Q. Du, and X. Yang, “Dog breed classification based on deep learning,” in 2020 13th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 2020.

[4] J. Du, “Understanding of object detection based on CNN family and YOLO,” J. Phys. Conf. Ser., vol. 1004, no. 2018, p. 012029, 2018.

[5] D. Jadhav and J. Singh, “A review on web information extraction and hidden predictive information from large databases,” Multimedia Tools and Applications, May 2025.

[6] D. Jadhav and J. Singh, “Web information extraction and fake news detection in twitter using optimized hybrid bi-gated deep learning network,” Multimedia Tools and Applications, May 2024.

[7] E. al. Jaibir Singh, “Enhancing Cloud Data Privacy with a Scalable Hybrid Approach: HE-DP-SMC,” Journal of Electrical Systems, vol. 19, no. 4, pp. 350–375, Jan. 2024.

[8] G. Sadineni, J. . Singh, S. . Rani, G. S. . Rao, M. J. . Pasha, and A. . Lavanya, “Blockchain-Enhanced Vehicular Ad-hoc Networks (B-VANETs): Decentralized Traffic Coordination and Anonymized Communication”, Int J Intell Syst Appl Eng, vol. 12, no. 1s, pp. 443–456, Sep. 2023.

[9] J. Singh, S. Rani, and G. Srilakshmi, “Towards Explainable AI: Interpretable Models for Complex Decision-making,” 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), pp. 1–5, Apr. 2024.

[10] J. Singh, S. Rani, and P. Kumar, “Blockchain and Smart Contracts: Evolution, Challenges, and Future Directions,” 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), pp. 1–5, Apr. 2024.

[11] J. Singh, S. Rani, and S. Devi, “Comment on ‘Prediction of Metabolic Dysfunction–Associated Steatotic Liver Disease via Advanced Machine Learning Among Chinese Han Population,’” Obesity Surgery, Sep. 2025.

[12] I. Ganie and S. Jagannathan, “Lifelong deep learning–based control of robot manipulators,” Int. J. Adapt. Control Signal Process., vol. 37, no. 12, pp. 3169–3192, 2023.

[13] I. Ganie and S. Jagannathan, “Lifelong learning control of nonlinear systems with constraints using multilayer neural networks with application to mobile robot tracking,” Proc. 2023 IEEE Conf. Control Technol. Appl. (CCTA), Bridgetown, Barbados, 2023, pp. 727–732.

[14] H. R. Laskar, “Adoption of fintech and digital financial services (DFS) by young professionals,” Int. J. Adv. Res. Eng. Technol., vol. 11, no. 1, pp. 537–561, 2020.

[15] H. R. Laskar, “Factors influencing saving and investment behavior of government and private sector employees,” Indian Journal of Economics and Business, vol. 20, no. 1, pp. 1168–1192, 2021.

[16] S. Roushon and H. R. Laskar, “Influence of Neural Behaviour on Decision Making,” IOSR Journal of Humanities and Social Science, vol. 29, no. 5, ser. 13, pp. 35–42, May 2024.

[17] Z. Alam and H. R. Laskar, “The Influence of Neural Behavior on Individuals’ Financial Decisions,” Journal of Economics, Finance and Management Studies, vol. 7, no. 6, pp. 3298–3306, Jun. 2024.

[18] A. Dhanai, H. S. Bagde, R. Gera, K. Mukherjee, C. Ghildiyal, and H. Yadav, “Case report on irritational fibroma,” Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S960–S962, Feb. 2024.

[19] H. Bagde, A. Dhopte, F. Bukhary, N. Momenah, F. Akhter, O. Mahmoud, K. P. Shetty, M. A. Shayeb, H. Abutayyem, and M. K. Alam, “Monkeypox and oral lesions associated with its occurrence: a systematic review and meta-analysis,” F1000Research, vol. 12, p. 964, Mar. 2024.

[20] H. Bagde, R. S. Karki, S. Husain, S. Khan, V. Haripriya, and P. Purwar, “Evaluation of microbiological flora in endo-perio lesions before and after treatment,” Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1707–S1709, Jun. 2025.

[21] B. Shyamsukha, H. Bagde, A. Sharan, M. Choudhary, A. Duble, and A. V. Dhan, “Evaluating the potential of ChatGPT as a supplementary intelligent virtual assistant in periodontology,” Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1415–S1417, Jun. 2025.

[22] H. S. Bagde, M. K. Alam, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, “The effect of a low-carbohydrate diet on periodontal health and inflammation in patients with type 2 diabetes,” Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S641–S643, Feb. 2024.
[23] H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, A. W. N. Alanazi, F. T. F. Alanazi, and M. G. Sghaireen, “The efficacy of platelet-rich plasma as an adjunct to bone grafting in alveolar ridge preservation following tooth extraction,” Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S564–S566, Feb. 2024.
[24] S. B. Mangalekar, H. S. Bagde, M. Sale, S. V. Jambhekar, C. Patil, and C. V. Deshmukh, “Comparing laser-assisted and conventional excision in the management of oral soft lesions: a prospective clinical study,” Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S859–S861, Feb. 2024.
[25] M. K. Alam, H. S. Bagde, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, “Comparing the long-term success rates of immediate implant placement vs delayed implant placement in patients with periodontally compromised teeth,” Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S626–S628, Feb. 2024.
[26] J. R. Rogers, Y. Wang, N. F. Khan, K. Mott, V. K. Nomula, D. Wang, P. C. Fiduccia, M. Burcu, and X. Liu, “Landscape assessment of clone-censor-weight methodology application in real-world data studies: A scoping review,” in Proceedings of the Pharmacoepidemiology and Drug Safety Conference, vol. 33, pp. 424–424, Nov. 1, 2024.
[27] I. A. Mohammed, "Artificial Intelligence in Supplier Selection and Performance Monitoring: A Framework for Supply Chain Managers," Educational Administration: Theory and Practice, vol. 29, no. 3, pp. 1186–1198, 2023.
[28] I. A. Mohammed, "The Role of Artificial Intelligence in Enhancing Business Efficiency and Supply Chain Management," Journal of Information Systems Engineering and Management, vol. 10, no. 10s, pp. 509–518, Feb. 2025.
[29] I. A. Mohammed, "AI-Powered Risk Management Frameworks for Ensuring Supplier Quality in Carbon Capture and Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 1, pp. 854–, Dec. 2023.
[30] I. A. Mohammed, "Optimizing Carbon Capture Supply Chains with AI-Driven Supplier Quality Management and Predictive Analytics," Journal of Next-Generation Research 5.0, Dec. 2024.
[31] I. A. Mohammed, "Machine Learning-Driven Predictive Models for Enhancing Supplier Reliability in Renewable Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, pp. 767–770, 2022.
[32] N. Gupta, M. Adawadkar, I. A. Mohammed, S. Verma, and M. Dubey, "Predictive Insights: Leveraging Artificial Intelligence for Strategic Business Decision-Making," Advances in Consumer Research, vol. 2, pp. 98–105, 2025.
[33] K. Chitra, S. S. Priscila, E. S. Soji, R. Rajpriya, B. Gayathri, and A. Chitra, “Transforming electrical simulation and management with smart grid technologies,” International Journal of Engineering Systems Modelling and Simulation, vol. 16, no. 4, pp. 241–253, 2025.
[34] M. V. Soosaimariyan, H. L. Allasi, K. Chitra, and J. B. Gnanadurai, “Enhanced EMG-based hand gesture recognition by using generalized deep infomax networks,” Journal of Sensors, vol. 2025, no. 1, p. 9496890, 2025.
[35] K. Lakshmi and K. Chitra, “Stress Net: Multimodal stress detection using ECG and EEG signals,” Journal of Data Science, vol. 2024, no. 59, pp. 1–8, 2024.
[36] S. Rishabh, K. Chitra, and C. S. Yap, “A study on non-fungible tokens marketplace for secure management,” INTI Journal, vol. 2024, no. 18, pp. 1–8, 2024.
[37] S. Shreyash, S. Gaur, K. Chitra, and M. Y. N. Zuhaili, “EasyLearnify – A student study portal,” INTI Journal, vol. 2024, no. 17, pp. 1–6, 2024.
[38] S. K. R. Padur, “Intelligent Resource Management: AI Methods for Predictive Workload Forecasting in Cloud Data Centers,” J. Artif. Intell. Mach. Learn. & Data Sci., vol. 1, no. 1, pp. 2936–2941, 2022.
[39] S. K. R. Padur, “Online Patching and Beyond: A Practical Blueprint for Oracle EBS R12.2 Upgrades,” Int. J. Sci. Res. Sci. Eng. Technol., vol. 2, no. 3, pp. 75–87, 2016.
[40] S. K. R. Padur, “Engineering Resilient Datacenter Migrations: Automation, Governance, and Hybrid Cloud Strategies,” Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 2, no. 1, pp. 340–348, 2017.
[41] S. K. R. Padur, “From Centralized Control to Democratized Insights: Migrating Enterprise Reporting from IBM Cognos to Microsoft Power BI,” Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 6, no. 1, pp. 218–225, 2020.

[42] S. K. Somayajula, "Enterprise Data Migration Success Patterns: Lessons from Large-Scale Transformations," International Journal of Research in Computer Applications and Information Technology (IJRCAIT), vol. 8, no. 1, pp. 757-776, Jan.-Feb. 2025.

[43] S. K. Somayajula, "Demystifying Modern Data Warehousing: From Traditional to Cloud-Native Solutions," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2025.

[44] S. K. Somayajula, "Building a Career in Enterprise Data Architecture: A Practical Guide," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), vol. 7, no. 1, Jan. 2025.

[45] S. K. Somayajula, "Advanced ETL Optimization: A Framework for Next-Generation Data Integration," International Journal of Computer Engineering and Technology (IJCET), vol. 16, no. 1, pp. 381-406, Jan.-Feb. 2025.

[46] S. Somayajula and A. Orlovsky, "Proof, Truth and Contradiction in the System and Meta-System: Comprehensive Mathematical Solutions and Implementation Framework," 2025.

[47] H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, K. K. Ganji, and M. G. Sghaireen, "Comparing the clinical and radiographic outcomes of two different surgical approaches for treating infrabony defects in chronic periodontitis patients," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S567–S569, Feb. 2024.

[48] A. Sharan, B. Pawar, H. Bagde, T. K. Chawla, A. V. Dhan, B. Shyamsukha, and S. Sharma, "Comparative evaluation of dentin hypersensitivity reduction over one month after a single topical application of three different materials: a prospective experimental study," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 4, pp. S3405–S3407, Dec. 2024.

[49] J. Prakash, T. Sinha, H. Bagde, N. Rajegowda, S. Bhat, A. Dhopte, M. Cicciù, and G. Minervini, "Evidence-based assessment of temporomandibular disorders in complete denture versus partial denture users: a systematic review," Minerva Dental and Oral Science, Sep. 2025.

[50] S. Laskar, H. R. Laskar, and M. N. I. Barbhuyan, "Perception of Women Entrepreneurs Regarding Social Media Marketing," Bangladesh Journal of Multidisciplinary Scientific Research, vol. 9, no. 5, pp. 10–18, Nov. 2024.

[51] I. Ganie and S. Jagannathan, "Lifelong learning-based multilayer neural network control of nonlinear continuous-time strict-feedback systems," Int. J. Robust Nonlinear Control, vol. 34, no. 2, pp. 1397–1416, 2023.

[52] J. Singh, S. Rani, and V. Kumar, "Role-Based Access Control (RBAC) Enabled Secure and Efficient Data Processing Framework for IoT Networks," International Journal of Communication Networks and Information Security (IJCNIS), Aug. 2024.

[53] J. Singh, S. Rani, S. Devi, and J. Kaur, "A Systematic Study on Recommendation System for E-Commerce Applications," 2025 Seventh International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 221–226, Apr. 2025.

[54] K. V. Deshpande and J. Singh, "A Systematic Review on Website Phishing Attack Detection for Online Users," International Journal of Image and Graphics, Jan. 2025.

[55] P. Nutalapati, "Automated Incident Response Using AI in Cloud Security," Journal of Artificial Intelligence, Machine Learning and Data Science, vol. 2, no. 1, pp. 1301–1311, 2024

[56] P. Nutalapati, "Data Leakage Prevention Strategies in Cloud Computing," European Journal of Advances in Engineering and Technology, vol. 8, no. 9, pp. 118–123, 2025.

[57] P. Nutalapati, "Security Considerations for Hybrid Cloud Deployments in Fintech Using Blockchain," Journal of Artificial Intelligence, Machine Learning and Data Science, vol. 1, no. 1, URF Publishers, 2025.

[58] P. Nutalapati, "Zero Trust Architecture in Cloud-Based Fintech Applications," Journal of Artificial Intelligence & Cloud Computing, vol. 2, no. 1, pp. 1–8, 2023.

[59] S. Arul Krishnan, V. P. Rameshkumaar, K. Prakash, and D. Sugandaran, "Impact of celebrity endorsement on cause-related marketing and purchase intention of FMCG consumers in Chennai," Journal of Statistics & Management Systems, vol. 26, no. 7, pp. 1627–1641, 2023.

[60] S. Arul Krishnan, V. P. Rameshkumaar, R. Sathya Aarthi, and S. Karthikeyan, "Role of job design on institutional support and work-life equilibrium of women teachers in Chennai," Journal of Statistics & Management Systems, vol. 26, no. 7, pp. 1549–1563, 2023.

[61] S. Karthikeyan, V. P. Rameshkumaar, and B. Balaji, "Subscribers of Indian Mobile Telecom: Satisfaction with Experience and Loyalty," International Journal of Management, vol. 11, no. 9, Oct. 15, 2020.

[62] S. Pundir, V. G. Murugan, P. Raman, V. P. Rameshkumaar, and P. Sudharsan, "Automatic Stock Price Prediction and Classification Based on Hybrid with AI Feature Selection Method," in Proc. 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), pp. 149–154, Apr. 9, 2024. IEEE.

[63] R. R. Chandan, S. Ramaraj, P. P. Devi, S. J. Sultanuddin, and E. Indhuma, "A Novel Machine Learning-Based Early Warning Detection System for Business Customer Churn," in 2024 5th International Conference on Recent Trends in Computer Science and Technology (ICRTCST), 2024, pp. 250-254, IEEE, Apr. 9, 2024.

[64] R. Sellappan, P. P. Devi, and C. R. Prabakar, "Service quality in retailing, banking and insurance: A comparative study," Global Management Review, vol. 7, no. 4, Aug. 1, 2013.

[65] T. Subashini and P. P. Devi, "Navigating the balance: Unveiling the factors influencing work-life harmony among teaching professionals," Indian Journal of Natural Sciences, vol. 16, no. 91, pp. 99157–99166, Aug. 2025.

[66] P. Nutalapati, J. R. Vummadi, S. Dodda and N. Kamuni, "Advancing Network Intrusion Detection: A Comparative Study of Clustering and Classification on NSL-KDD Data," 2025 International Conference on Data Science and Its Applications (ICoDSA), Jakarta, Indonesia, 2025, pp. 880-885.

[67] P. Nutalapati, S. M. Dhavale, A. Shrivastava, R. V. S. Praveen, H. K. Vemuri and R. RiadHwsein, "IoT and Machine Learning-Enhanced Energy Management in Enabled Smart Grids for Predictive Load Balancing," 2025 World Skills Conference on Universal Data Analytics and Sciences (WorldSUAS), Indore, India, 2025.

[68] P. Nutralapati, “Disaster Recovery and Business Continuity Planning in Cloud-Blockchain Infrastructures,” SSRN Electron. J., Jun. 2020. Available: https://ssrn.com/abstract=5219038.

[69] S. G. K. Peddireddy, “Advancing Threat Detection in Cybersecurity through Deep Learning Algorithms,” FMDB Transactions on Sustainable Intelligent Networks., vol.1, no. 4, pp. 190–200, 2024.

[70] S. G. K. Peddireddy, “Integrating AI for Proactive Network Defense against Emerging Security Vulnerabilities,” FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 232–241, 2024.

[71] S. G. K. Peddireddy, “Optimizing Resource Allocation in Multi-Cloud Environments for Cost Efficiency and Scalability,” FMDB Transactions on Sustainable Computing Systems., vol. 2, no. 4, pp. 167–177, 2024.

[72] V. Hiremath, “Quantum Networking: Strategic Imperatives for Enterprises and Service Providers in the Emerging Quantum Era,” Journal of Computational Analysis and Applications (JoCAAA), vol. 31, no. 3, pp. 617–631, Dec. 2023.

[73] V. Hiremath, "AI-Optimized Adaptive Routing for High-Performance Data Centers: A Predictive Framework for Dynamic Network Optimization," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 152-157.

[74] V. Hiremath, "Optimizing SDN Controller Placement for Enhanced Performance and Scalability in Large-Scale Networks," 2025 International Conference on Intelligent Communication Networks and Computational Techniques (ICICNCT), Bidar, India, 2025.

[75] D. K. Arora et al., “An in vitro assessment of microleakage of pit and fissure sealants and restorative materials using dye penetration method,” Journal of Pharmacy and Bioallied Sciences, Feb. 2025.

[76] R. Nagar et al., “In vitro analysis of compressive strength of three different aesthetic restorative materials,” Journal of Pharmacy and Bioallied Sciences, Feb. 2025.

[77] N. Maiti et al., “Assessment of the efficacy of photobiomodulation (PBM) therapy in periodontal treatment: a longitudinal study,” Journal of Pharmacy and Bioallied Sciences, vol. 16, no. Suppl 3, pp. S2449–S2451, Jul. 2024.

[78] D. Sumathi and P. Poongodi, "Scheduling Based on Hybrid Particle Swarm Optimization with Cuckoo Search Algorithm in Cloud Environment," IIOAB Journal, vol. 7, no. 9, pp. 358-366, 2016.

[79] D. Sumathi and P. Poongodi, "Secure medical information processing in cloud: Trust with swarm based scheduling," Journal of Medical Imaging and Health Informatics, vol. 6, no. 7, pp. 1636-1640, 2016.

[80] D. Sumathi and P. Poongodi, "An improved scheduling strategy in cloud using trust based mechanism," Int. J. Comput. Electr. Autom. Control Inf. Eng, vol. 9, no. 2, pp. 637-641, 2015.

[81] D. Sumathi, B. Melinamath, and R. Goyal, "Iov Traffic Prediction Utilizing Bidirectional Memory and Spatiotemporal Constraints with Local Search and NonLinear Analysis," Journal of Computational Analysis & Applications, vol. 33, no. 2, 2024.

[82] D. Sumathi, A. Singh, A. Sinha, D. Aditya, and M. R. KF, "The Deepfake Dilemma: Enhancing Deepfake Detection with Vision Transformers," in 2025 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics, Jan. 2025.

[83] V. B. Gowda, M. T. Gopalakrishna, J. Megha, and S. Mohankumar, “Foreground segmentation network using transposed convolutional neural networks and up sampling for multiscale feature encoding,” Neural Netw., vol. 170, pp. 167–175, 2024.

[84] V. B. Gowda, G. M. Thimmaiah, M. Jaishankar, and C. Y. Lokkondra, "Background-foreground segmentation using Multi-scale Attention Net (MA-Net): A deep learning approach," Rev. Intell. Artif., vol. 37, no. 3, pp. 557–565, 2023.
[85] V. B. Gowda, M. G. Krishna, and J. Megha, "Dynamic Background Modeling and Foreground Detection using Orthogonal Projection onto the Subspace of Moving Objects," in Proc. IC3, 2023.
[86] V. B. Gowda, M. T. Gopalakrishna, J. Megha, and S. Mohankumar, "Background initialization in video data using singular value decomposition and robust principal component analysis," Int. J. Comput. Appl., vol. 45, no. 9, pp. 600–609, 2023.
[87] A. K. Joshi and S. B. Kulkarni, "Flow analysis of vehicles on a lane using deep learning techniques," J. Adv. Inf. Technol., vol. 14, no. 6, pp. 1354–1364, 2023.
[88] A. K. Joshi, V. Shirol, S. Jogar, P. Naik, and A. Yaligar, "Credit card fraud detection using machine learning techniques," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 6, no. 3, pp. 436–442, 2020.
[89] A. K. Joshi and S. B. Kulkarni, "Multi-modal information fusion for localization of emergency vehicles," Int. J. Image Graph., vol. 24, no. 1, Art. no. 2550050, 2024.
[90] A. K. Joshi and S. B. Kulkarni, "Multimodal deep learning information fusion for fine-grained traffic state estimation and intelligent traffic control," Int. J. Intell. Syst. Appl. Eng., vol. 11, no. 3, pp. 1020–1029, 2023.
[91] V. S. A. Anala, A. R. Pothu, and S. Chintapalli, "Enhancing Preventive Healthcare with Wearable Health Technology for Early Intervention," FMDB Transactions on Sustainable Health Science Letters., vol.2, no.4, pp. 211–220, 2024.
[92] V. S. A. Anala and S. Chintapalli, "Scalable Data Partitioning Strategies for Efficient Query Optimization in Cloud Data Warehouses," FMDB Transactions on Sustainable Computer Letters., vol. 2, no. 4, pp. 195–206, 2024.
[93] S. N. Akhter, R. Kumari, and A. Kumar, "Fertility booster effect of Asparagus recemosus against arsenic induced reproductive toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.
[94] Z. Hashmi, R. Kumari, and A. Kumar, "Antidote effect of Bacopa Koneru against arsenic induced toxicity in rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.
[95] Z. Hashmi, R. Kumari, and A. Kumar, "Phytoremedial effect of Ocimum sanctum against arsenic induced toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com. ISSN: 0253-7214.
[96] B. Kumari, P. Das, and R. Kumari, "Accelerated processing of solitary and clustered abasic site DNA damage lesion by APE1 in the presence of aqueous extract of Ganoderma lucidum," J. Biosci., vol. 41, pp. 265–275, 2016.
[97] R. Kumari, R. K. Singh, N. Kumar, and R. Kumari, "Preparation of superfine Bael leaf nanopowder, physical properties measurements and its antimicrobial activities," Egypt. Chem. Bull., vol. 12, no. 4, pp. 284–297, 2023.
[98] M. K. Sinha, R. Kumari, and A. Kumar, "Ameliorative effect of Ganoderma lucidum on sodium arsenite induced toxicity in Charles Foster rats," J. Adv. Zool., vol. 45, no. 5, 2024. [Online]. Available: http://jazindia.com.