# Enhanced Classification of Reptiles and Amphibians Using Deep Learning with Ensemble and Data Augmentation

**M. Koushik Anand Sai[1], N. Nageshwar Rao[2]**

[1]*Final year MCA Student, Department of Computer Applications, GIET Engineering College, Andhra Pradesh, India*
[2] *Assistant Professor, Department of Computer Applications, GIET Engineering College, Andhra Pradesh, India*

---

**Abstract-***This project presents an image-based classification system for identifying reptiles and amphibians using deep learning. The system is designed to help users upload an animal image and receive a predicted class along with its biological category, such as reptile or amphibian. A convolutional neural network model based on MobileNetV2 is used because it provides a good balance between accuracy and performance, making it suitable for practical web-based deployment. The dataset contains multiple animal classes including chameleon, frog, gecko, iguana, snake, toad, salamander, crocodile/alligator, lizard, and turtle/tortoise. The application is developed using Django for the backend and a clean web interface for user interaction. Users can register, log in, train the model, and perform image prediction through the browser. The admin module manages registered users and controls account activation. The trained model processes uploaded images by resizing and preprocessing them before generating predictions. The system also displays confidence values, helping users understand how reliable each prediction is. Overall, this project combines machine learning and web technology to create a simple, accessible, and useful classification platform. It demonstrates how deep learning can be applied to biological image recognition and provides a foundation for future improvements such as larger datasets, better model tuning, mobile support, and real-time classification.*

**Key Words: Reptile Classification, Amphibian Classification, Deep Learning, Image Classification, Convolutional Neural Network, MobileNetV2, Django Web Application, Animal Recognition, Computer Vision, Machine Learning, Image Processing, Species Identification, Transfer Learning, Prediction System.**

## 1.INTRODUCTION

The identification of reptiles and amphibians is an important task in biology, education, wildlife observation, and environmental study. Many species share similar body structures, colours, and patterns, which can make manual identification difficult for students, researchers, and general users. With the growth of artificial intelligence and computer vision, image-based classification has become a practical way to recognise living organisms more quickly and accurately.

This project focuses on developing a web-based system for classifying reptiles and amphibians from uploaded images. The system uses a deep learning model trained on different animal classes such as chameleon, frog, gecko, iguana, snake, toad, salamander, crocodile/alligator, lizard, and turtle/tortoise. When a user uploads an image, the model analyses its visual features and predicts the most suitable animal class along with its category as reptile or amphibian.

The application is built using Django, which provides a structured backend for user registration, login, admin control, model training, and prediction. A clean web interface allows users to interact with the system easily without needing technical knowledge of machine learning. The model is based on MobileNetV2, a lightweight convolutional neural network architecture known for efficient image classification. This makes the system suitable for practical use while maintaining good prediction performance.

The main aim of this project is to combine deep learning with a user-friendly web platform to support animal image recognition. It reduces the effort needed for manual classification and provides quick prediction results with confidence scores. This project also shows how machine learning can be applied in the field of wildlife identification and can be extended in the future with larger datasets, improved accuracy, and real-time mobile-based classification.

## 2. METHODOLOGY

The methodology of this project follows a step-by-step process that includes dataset collection, image preprocessing, model training, web application development, and prediction output generation. The main purpose of this method is to build a system that can classify reptiles and amphibians from uploaded images through a simple web interface.

First, an image dataset containing different reptile and amphibian classes was prepared. The dataset includes classes such as chameleon, crocodile/alligator, frog, gecko, iguana, lizard, salamander, snake, toad, and turtle/tortoise. The images were arranged into separate folders according to their class names so that the model could learn the visual differences between each category.

After preparing the dataset, image preprocessing was performed before training. Each image was resized to a fixed size of `224 x 224` pixels, which is suitable for the MobileNetV2 model. The images were also converted into numerical arrays and normalized using the required preprocessing method. This step helps the model understand image features more effectively and improves training consistency.

The dataset was divided into training and validation parts. The training data was used to teach the model, while the validation data was used to check how well the model performed on unseen images during training. Data augmentation techniques such as rotation, zooming, shifting, and horizontal flipping were applied to increase image variation. This helps reduce overfitting and improves the model's ability to handle real-world images.

For classification, the MobileNetV2 deep learning architecture was used. MobileNetV2 is a lightweight convolutional neural network that is commonly used for image classification tasks. The base model was used to extract important image features, and additional dense layers were added to classify the images into the selected animal classes. The model was trained using categorical cross-entropy loss and an Adam optimizer.

Once training was completed, the trained model was saved and connected to the Django web application. The backend of the application handles user registration, login, admin approval, model training, and prediction requests. When a user uploads an image, the system stores the image temporarily, preprocesses it in the same way as the training images, and sends it to the trained model for prediction.

The model then generates a probability score for each class. The class with the highest probability is selected as the final prediction. The system also maps the predicted animal class to its broader biological category, such as reptile or amphibian. Finally, the predicted animal name, category, and confidence percentage are displayed on the web page for the user.

This methodology provides a complete workflow from raw image data to final web-based prediction. By combining deep learning with Django, the project creates a practical classification system that can be used easily by users without requiring direct knowledge of machine learning or programming.

## 2.1 Data Collection and Preprocessing

Data collection is one of the most important stages of this project because the performance of the classification model depends strongly on the quality of the images used for training. The dataset used in this project contains images of reptiles and amphibians arranged into separate class folders. Each folder represents one animal class, such as chameleon, crocodile/alligator, frog, gecko, iguana, lizard, salamander, snake, toad, and turtle/tortoise.

The images were collected and organized in a structured format so that the model could learn each class separately. This folder-based arrangement helps the training process because the class name is directly taken from the folder name. By using labelled folders, the system can understand which image belongs to which animal category during training.

Before training the model, all images are passed through preprocessing steps. Since image sizes may differ from one file to another, every image is resized to `224 x 224` pixels. This size is required because the MobileNetV2 model expects input images in a fixed dimension. Resizing also helps maintain consistency throughout the training and prediction process.

After resizing, the images are converted into numerical array format. A deep learning model cannot directly understand an image as humans see it, so each image must be represented as pixel values. These pixel values are then normalized using the MobileNetV2 preprocessing function. Normalization helps bring the image data into a suitable range, allowing the model to learn patterns more effectively.

The dataset is divided into training and validation data. The training data is used to teach the model, while the validation data is used to check the model's performance on images it has not directly learned from. This separation helps identify whether the model is learning properly or simply memorizing the training images.

Data augmentation is also applied during training to improve the model's ability to handle variations. Techniques such as rotation, zooming, shifting, and horizontal flipping are used to create different versions of the same image. These changes help the model become more flexible when dealing with real-world images taken from different angles, distances, and lighting conditions.

During prediction, the uploaded image goes through the same preprocessing steps as the training images. It is resized, converted into an array, normalized, and then passed to the trained model. Using the same preprocessing method during both training and prediction helps maintain accuracy and consistency.

Overall, data collection and preprocessing provide the foundation for the entire classification system. A properly organized dataset and consistent preprocessing pipeline allow the model to learn meaningful visual features and produce more reliable predictions for reptile and amphibian images.

## 2.2 The Core Classification Model (MobileNetV2)

The core classification model used in this project is based on MobileNetV2, a convolutional neural network architecture designed for efficient image classification. It is selected because it can learn useful visual features from images while remaining lighter and faster than many large deep learning models. This makes it suitable for a web-based application where prediction speed and system performance are important.

In this project, MobileNetV2 acts as the main feature extraction model. When an image is given to the system, the model studies important visual patterns such as shape, colour, texture, edges, and body structure. These features help the system understand the difference between reptiles and amphibians, as well as the individual animal classes present in the dataset.

The dataset contains multiple classes, including chameleon, crocodile/alligator, frog, gecko, iguana, lizard, salamander, snake, toad, and turtle/tortoise. During training, the model learns from these labelled image folders and gradually improves its ability to separate one class from another. Each image is resized to `224 x 224` pixels before being passed into the model, which matches the expected input size of MobileNetV2.

The original MobileNetV2 base is used to extract image features, and additional dense layers are added at the end for classification. These final layers are responsible for converting the learned features into class probabilities. The output layer uses a softmax activation function, which gives a probability score for each animal class. The class with the highest probability is selected as the final prediction.

The model is trained using the Adam optimizer and categorical cross-entropy loss function. The Adam optimizer helps update the model weights efficiently during training, while categorical cross-entropy measures how far the predicted class is from the actual class. The training process also uses validation data to monitor performance and reduce overfitting.

After training, the model is saved and connected with the Django web application. During prediction, an uploaded image is pre-processed in the same way as the training images and then passed to the saved model. The model returns the predicted class and confidence value, which are displayed to the user along with the broader category, such as reptile or amphibian.

Overall, the core classification model forms the intelligence of the project. It allows the system to move beyond simple image uploading and provides meaningful identification based on learned visual features. By using MobileNetV2, the project achieves a balance between accuracy, speed, and practical usability.

## 2.3 The Styling Recommendation Engine

The styling recommendation engine in this project is designed to improve the way prediction results are presented to the user. After the model classifies an uploaded image, the system does not simply display the animal name; it also organizes the output in a clear and understandable format. This helps the user quickly identify whether the predicted animal belongs to the reptile or amphibian category.

The engine supports the visual presentation of results by using different layout elements, colors, and confidence-based information. For example, when the system predicts an animal such as a snake, lizard, or turtle, the result is shown under the reptile category. Similarly, animals such as frogs, toads, and salamanders are shown under the amphibian category. This separation makes the output easier to read and more meaningful for users.

The recommendation part of the engine focuses on guiding the user based on the prediction confidence. If the model gives a high confidence value, the result can be considered more reliable. If the confidence value is low, the system helps the user understand that the uploaded image may not be clear enough or may contain background disturbance. In this way, the system provides not only a prediction but also useful feedback about the quality of the result.

The styling engine also improves the overall user experience of the web application. The prediction page, training page, login page, and dashboard are arranged with a clean and simple design so that users can navigate without confusion. Important actions such as uploading an image, training the model, and viewing results are placed clearly on the interface. This makes the system suitable even for users who do not have technical knowledge.

Overall, the styling recommendation engine acts as a bridge between the deep learning model and the user. It converts technical prediction output into a more readable and user-friendly result. By combining classification results with proper visual presentation and confidence-based guidance, the system becomes more practical, understandable, and effective for reptile and amphibian identification.

## 2.4 System Integration and Deployment

The system integration stage connects the trained deep learning model with the Django web application so that users can access the classification process through a browser. The project is divided into different modules such as user registration, user login, admin login, model training, and image prediction. Each module works together to provide a complete workflow from user access to final classification output.

The backend of the system is developed using Django. It manages URL routing, database operations, user details, admin actions, file upload handling, and communication with the machine learning model. The SQLite database is used to store registered user information, login details, account status, and other required records. The admin module allows the administrator to view registered users, activate accounts, and remove users when needed.

The trained MobileNetV2 model is saved as a model file and integrated into the prediction module. When a user uploads an image, the Django backend receives the file, stores it in the media directory, and sends it for preprocessing. The image is resized, converted into an array, normalized, and passed to the trained model. The model then returns the predicted class and confidence score, which are displayed on the prediction result page.

The frontend is connected with the backend through Django templates. Pages such as home, registration, login, training, and prediction are rendered using HTML and CSS. Forms are used to collect user details and image uploads, while Django views process the submitted data. This integration allows the user to interact with the system without needing to know how the machine learning model works internally.

During deployment, the required Python packages are installed inside a virtual environment. Django, TensorFlow, NumPy, scikit-learn, Matplotlib, Seaborn, and other dependencies are configured before running the project. The database migrations are applied to create the necessary tables. After setup, the Django development server is started, and the system can be accessed locally through a web browser.

The deployment process also includes placing the dataset, trained model, static files, and media files in their correct directories. Static files are used for styling and frontend design, while media files are used for uploaded images and dataset storage. The trained model file is loaded during prediction, so it must be available in the project root directory.

For local deployment, the project can be run using the command `python manage.py runserver. Once the server starts, users can open the application using the localhost address. The system is mainly designed for local or academic demonstration, but it can be further deployed on a cloud server by configuring production settings, allowed hosts, static file handling, database settings, and security options.

Overall, the integration and deployment process ensures that the machine learning model, web interface, database, and user modules work as a single complete application. This makes the project practical, easy to access, and suitable for demonstrating reptile and amphibian classification through a web-based platform.

## 3. DISCUSSION

The developed system shows how deep learning can be used to classify reptiles and amphibians through image input in a simple web-based environment. Instead of depending only on manual observation, the system studies the visual features present in an uploaded image and gives a predicted animal class with a confidence value. This makes the project useful for students, beginners, and users who may not have detailed knowledge about different species.

One of the important points observed in this project is that reptiles and amphibians can sometimes have similar shapes, skin textures, and body colours. Because of this, classification is not always straightforward. For example, some lizards and salamanders may look similar in certain images, especially when the image quality is low or the background is unclear. The model performs better when the uploaded image clearly shows the animal and contains less background disturbance.

The use of MobileNetV2 is suitable for this project because it is lightweight and efficient compared to many larger deep learning models. It helps reduce training and prediction time while still giving good classification performance. Since the project is connected with a Django web application, users can access the model through a browser without directly handling the machine learning code. This makes the system more practical and easier to use.

The admin and user modules also improve the usability of the application. User registration and login provide controlled access, while the admin can activate users and manage registered accounts. The training and prediction modules are separated clearly, which makes the system easier to understand and maintain. The prediction result, along with the confidence percentage, gives users a better idea of how strongly the model supports its output.

However, the accuracy of the system depends greatly on the quality and variety of the dataset. If the dataset contains limited images, repeated backgrounds, or uneven class distribution, the model may become biased toward certain classes. For better real-world performance, the dataset should include more images taken from different angles, lighting conditions, backgrounds, and species variations. Future improvements can also include real-time camera-based prediction, mobile support, larger datasets, and advanced model fine-tuning.

Overall, this project demonstrates a meaningful use of artificial intelligence in animal classification. It connects computer vision with biological identification and provides a foundation for further development in wildlife recognition systems. The project is not only useful as a technical implementation but also shows how machine learning can support learning, research, and awareness in the field of reptiles and amphibians.

## 4. CONCLUSIONS

This project successfully demonstrates how deep learning can be used to classify reptiles and amphibians through image-based prediction. By using a trained MobileNetV2 model, the system is able to analyse uploaded animal images and provide a predicted class along with its category and confidence value. The project brings together computer vision and web development in a practical way, making the classification process simple for users who may not have technical knowledge.

The Django-based web application provides important features such as user registration, login, admin approval, model training, and image prediction. These features make the system more complete and easier to manage. The clean interface allows users to upload images and view results without difficulty, while the backend handles model loading, preprocessing, and prediction.

Overall, the project shows that artificial intelligence can support animal identification tasks in an efficient and accessible manner. It can be useful for students, researchers, and wildlife enthusiasts who need quick assistance in recognizing reptiles and amphibians. In the future, the system can be improved by adding more species, increasing the dataset size, improving model accuracy, and supporting real-time prediction through mobile or camera-based input.

## REFERENCES

[1] Tan, W.C., Herrel, A. and Rödder, D., 2023. A global analysis of habitat fragmentation research in reptiles and amphibians: what have we done so far?. Biodiversity and Conservation, 32(2), pp.439-468.

[2] Gill, K.S., Anand, V. and Gupta, R., 2023, August. An Efficient VGG19 Framework for Malaria Detection in Blood Cell Images. In 2023 3rd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-4). IEEE.

[3] Fraser, M., 2023. Reptiles and Amphibians of the Cape of Good Hope Nature Reserve, Western Cape, South Africa. Biodiversity Observations, 13, pp.162-185.

[4] Banerjee, D., Kukreja, V., Hariharan, S., Jain, V. and Dutta, S., 2023, May. Hybrid CNN-SVM Approach with Regularization for Accurate Classification of Images: A Case Study on Rudraksha Beads. In 2023 International Conference on Disruptive Technologies (ICDT) (pp. 676-680). IEEE.

[5] Abdullah, S.S., Dela Cruz, K.C., Casim, L.F., Agduma, A.R. and Tanalgo, K.C., 2023. Leaping Forward or Crawling Backward? Efforts and Biases in Amphibian and Reptile Research on a Megadiverse Faunal Region in the Philippines. Conservation, 3(3), pp.363-378.

[6] Gill, K.S., Anand, V. and Gupta, R., 2023, August. An Intelligent System for Dysarthria Classification of Male and Female Processed Dataset using Sequential Model Parameters. In 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS) (pp. 815-820). IEEE.

[7] Inman, R.D., Esque, T.C. and Nussear, K.E., 2023. Dispersal limitations increase vulnerability under climate change for reptiles and amphibians in the southwestern United States. The Journal of Wildlife Management, 87(1), p.e22317.

[8] Reshan, M.S.A., Gill, K.S., Anand, V., Gupta, S., Alshahrani, H., Sulaiman, A. and Shaikh, A., 2023, May. Detection of Pneumonia from Chest X-ray Images Utilizing MobileNet Model. In Healthcare (Vol. 11, No. 11, p. 1561). MDPI.

[9] Schilliger, L., Paillusseau, C., François, C. and Bonwitt, J., 2023. Major Emerging Fungal Diseases of Reptiles and Amphibians. Pathogens, 12(3), p.429.

[10] Banerjee, D., Kukreja, V., Hariharan, S., Jain, V. and Dutta, S., 2023, May. CNN-SVM Model for Accurate Detection of Bacterial Diseases in Cucumber Leaves. In 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 7-12). IEEE.

[11] Vaz-Canosa, P., Laufer, G., Borteiro, C., Baldo, D., Prigioni, C. and Soutullo, A., 2023. Expert-based assessment of the climate change vulnerability of amphibians and reptiles of Uruguay. Environmental Conservation, 50(1), pp.12-21.

[12] Gill, K.S., Sharma, A., Anand, V. and Gupta, R., 2023, May. Smart Shoe Classification Using Artificial Intelligence on EfficientnetB3 Model. In 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT) (pp. 254-258). IEEE.

[13] Aburrow, K., Moffat, D., Bega, S. and Clarke, K., 2023. What is wrong with wildlife fencing and what should we do? A review of fencing guidance for reptiles and amphibians.

[14] Banerjee, D., Kukreja, V., Gupta, A., Singh, V. and Brar, T.P.S., 2023, August. Combining CNN and SVM for Accurate Identification of Ridge Gourd Leaf Diseases. In 2023 3rd Asian Conference on Innovation in Technology (ASIANCON) (pp. 1-5). IEEE.

[15] Clarke, S., Jackson, M. and Badal, S., 2024. Animal metabolites: from amphibians, reptiles, Aves/birds, and invertebrates. In Pharmacognosy (pp. 441-454). Academic Press.

[16] Gill, K.S., Sharma, A., Anand, V. and Gupta, R., 2023, January. Categorization of Butterflies using Convolutional Neural Network. In 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT) (pp. 1-5). IEEE.

[17] Griffin, R.K., Tzanopoulos, J. and Griffiths, R.A., 2023. Comparison of national and regional status assessments of amphibians and reptiles in Guatemala. Herpetological Review.

[18] Banerjee, D., Kukreja, V., Sharma, V., Jain, V. and Hariharan, S., 2023, June. Automated Diagnosis of Marigold Leaf Diseases using a Hybrid CNN-SVM Model. In 2023 8th International Conference on Communication and Electronics Systems (ICCES) (pp. 901-906). IEEE.

[19] Marsh, D.M., Medina, J. and Wilkinson, B., 2023. Predicting amphibian and reptile distributions and species richness across urban parks in the eastern United States. Urban Ecosystems, pp.1-21.

[20] Gill, K.S. and Gupta, R., 2023, May. Chronic Kidney Disease Detection Using GridSearchCV Cross Validation Method. In 2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON) (pp. 318-322). IEEE.