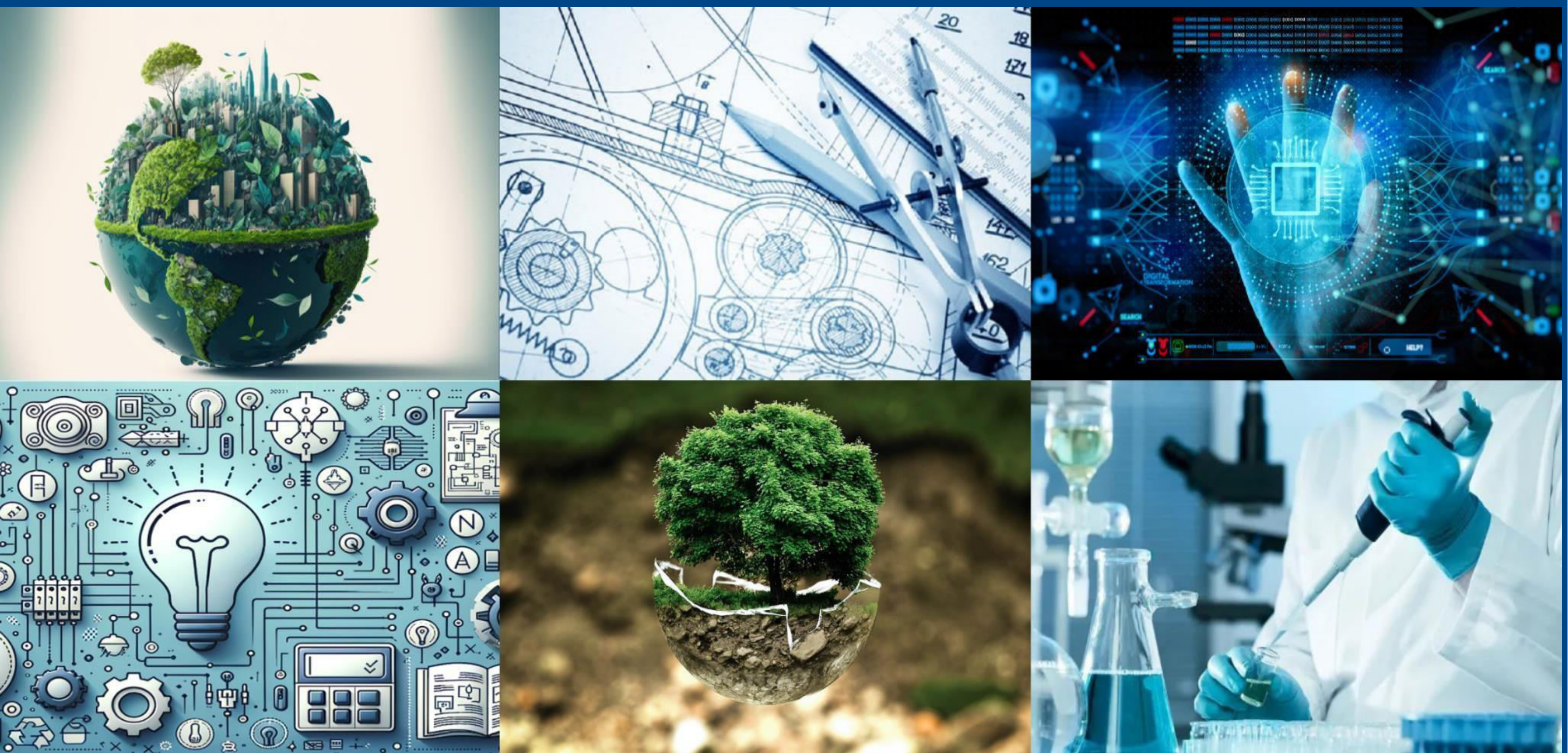
ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 8, Issue 7, July 2025

www.ijmrset.com ijmrset@gmail.com +91 6381 907 438 +91 63819 07438

© 2025 IJMRSET | Volume 8, Issue 7, July 2025| DOI:10.15680/IJMRSET.2025.0807044

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Advanced CNN-Based Deep Learning for Accurate Cannie Skin Disease Diagnosis

Mageshwari M[1], Harini K[2], Lalithambigai S[3], Thenmozhi V[4], Sudarsan K[5]

Assistant Professor, Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering and Technology, Chengalpattu, Tamil Nadu, India[1]

UG Scholars, Department of Computer Science and Engineering, Karpaga Vinayaga College of Engineering and Technology, Chengalpattu, Tamil Nadu, India[2, 3, 4, 5]

ABSTRACT: Accurate diagnosis of skin diseases in dogs is essential for timely treatment, but traditional methods rely heavily on veterinary expertise, making them subjective and time-consuming. This study explores the use of deep learning, specifically Convolutional Neural Networks (CNNs), to automate the diagnosis of canine skin conditions. A CNN-based system is trained on a dataset of dog skin images covering common disorders like dermatitis, mange, hot spots, and infections. To enhance usability, an interactive chatbot is integrated, allowing pet owners and veterinarians to input symptoms and receive preliminary diagnoses or recommendations for professional consultation. The CNN model is fine-tuned for classification and evaluated using accuracy metrics. Our findings highlight the potential of CNNs to distinguish between healthy and diseased skin effectively, while the chatbot improves accessibility and user engagement. This research advances AI-driven veterinary dermatology, providing a scalable and reliable solution for early detection and management of canine skin diseases.

KEYWORDS: Convolutional Neural Networks (CNN), dog skin disease detection, deep learning, veterinary dermatology, image classification, chatbot integration, early diagnosis, canine health, AI in veterinary medicine, automated disease identification.

I. INTRODUCTION

Skin diseases in dogs are a prevalent concern among pet owners and veterinarians, as they can significantly affect a dog's health and quality of life. These conditions can range from common allergies and infections to more severe disorders such as mange, fungal infections, and even skin cancer. Traditional methods of diagnosis involve visual inspection by veterinarians, followed by laboratory tests such as skin scrapings, biopsies, or microbial cultures. However, these approaches can be time-consuming, costly, and sometimes lead to misdiagnoses due to human error or overlapping symptoms among different conditions.With recent advancements in artificial intelligence (AI) and deep learning, particularly Convolutional Neural Networks (CNNs), automated and highly accurate diagnostic systems have become feasible for veterinary applications. CNNs have demonstrated exceptional performance in image classification tasks, making them an ideal approach for diagnosing skin diseases based on high-resolution images of affected areas. By leveraging deep learning techniques, this project aims to develop a robust and efficient model capable of identifying and classifying various canine skin conditions with high accuracy.The proposed system utilizes a CNN-based architecture trained on a diverse dataset of dog skin diseaseimages. The model processes input images, extracts essential features, and categorizes skin conditions based on learned patterns. Additionally, integrating this deep learning approach into a web or mobile application can allow pet owners and veterinarians to obtain rapid and accurate preliminary diagnoses, aiding in early detection and timely treatment.This research not only contributes to the field of veterinary medicine but also highlights the potential of AI in improving animal healthcare. By minimizing diagnostic errors and expediting the identification of skin diseases, deep learning-based solutions can enhance treatment outcomes and overall pet well-being.

1.1 Data Science:

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from structured and unstructured data, and apply knowledge and actionable insights from data across a broad range of application domains.

© 2025 IJMRSET | Volume 8, Issue 7, July 2025| DOI:10.15680/IJMRSET.2025.0807044

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

The term "data science" was first coined in 2008 by D.J. Patil, and Jeff Hammer bacher, the pioneer leads of data and analytics efforts at LinkedIn and Facebook. In less than a decade, it has become one of the hottest and most trending professions in the market.

Data science is the field of study that combines domain expertise, programming skills, and knowledge of mathematics and statistics to extract meaningful insights from data.

Data science can be defined as a blend of mathematics, business acumen, tools, algorithms and machine learning techniques, all of which help us in finding out the hidden insights or patterns from raw data which can be of major use in the formation of big business decisions.

Data Scientist:
Data scientists examine which questions need answering and where to find the related data. They have business acumen and analytical skills as well as the ability to mine, clean, and present data. Businesses use data scientists to source, manage, and analyze large amounts of unstructured data.

II. METHODOLOGY

The project methodology for the Skin Disease Classification and using Artificial Intelligence Techniques is designed to systematically guide the development and implementation of a robust system for identifying and categorizing skin lesions. The primary goal is to leverage state- of-the-art artificial intelligence (AI) techniques to enhance the accuracy and efficiency of skin cancer diagnosis. This methodology encompasses various stages, including data collection, pre-processing, model development, training, and evaluation. The integration of AI techniques aims to empower medical professionals with a tool that can assist in early and accurate detection of Skin Disease , thereby improving patient outcomes.

The train dataset is used to train the model (CNN) so that it can identify the test image and the disease it has CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D. After the model is trained successfully, the software can identify the Skin Disease and Non SkinDisease Classification image contained in the dataset. After successful training and preprocessing, comparison of the test image and trained model takes place to predict.

III. ARCHITECTURE OF CNN CONVOLUTIONAL NEURAL NETWORK

A Convolutional neural network (CNN) is one type of Artificial Neural Network. A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, and also for other auto correlated data.

Models API:
There are three ways to create Keras models:
- The Sequential model, which is very straightforward (a simple list of layers), but is limited to single-input, single-output stacks of layers (as the name gives away).
- The Functional API, which is an easy-to-use, fully- featured API that supports arbitrary model architectures. For most people and most use cases, this is what you should be using. This is the Keras "industry strength" model.
- Model subclassing, where you implement everything from scratch on your own. Use this if you have complex, out-of-the-box research use cases.

Types of Keras Models
Models in keras are available in two types:
- Keras Sequential Model
- Keras Functional API

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

1. Sequential Model in Keras

It allows us to create models layer by layer in sequential order. But it does not allow us to create models that have multiple inputs or outputs. It is best for simple stack of layers which have 1 input tensor and 1 output tensor. This model is not suited when any of the layer in the stack has multiple inputs or outputs. Even if we want non-linear topology, it is not suited.

2. Functional API in Keras

It provides more flexibility to define a model and add layers in keras. Functional API allows us to create models that have multiple input or output. It also allows us to share these layers. In other words. we can make graphs of layers using Keras functional API. As functional API is a data structure, it is easy to save it as a single file that helps in recreating the exact model without having the original code. Also its easy to model the graph here and access its nodes as well.

3. Model Subclassing in Keras

Sequential model does not allow you much flexibility to create your models. Functional API also only has a little of customization available for you. But you may create your own fully-customizable models in Keras. This is done by subclassing the Model class and implementing a call method. Input() is used to instantiate a Keras tensor. A Keras tensor is a symbolic tensor-like object, which we augment with certain attributes that allow us to build a Keras model just by knowing the inputs and outputs of the model. For instance, if a, b and c are Keras tensors, it becomes possible to do: model = Model(input=[a, b], output=c)

kernels: ⟶

Each convolutional layer contains a series of filters known as convolutional kernels. The filter is a matrix of integers that are used on a subset of the input pixel values, the same size as the kernel. Each pixel is multiplied by the corresponding value in the kernel, then the result is summed up for a single value for simplicity representing a grid cell, like a pixel, in the output channel/feature map. These are linear transformations, each convolution is a type of affine function.

In computer vision the input is often a 3 channel RGB image. For simplicity, if we take a greyscale image that has one channel (a two dimensional matrix) and a 3x3 convolutional kernel (a two dimensional matrix). The kernel strides over the input matrix of numbers moving horizontally column by column, sliding/scanning over the first rows in the matrix containing the images pixel values. Then the kernel strides down vertically to subsequent rows. Note, the filter may stride over one or several pixels at a time, this is detailed further below. In other non-vision applications, a one dimensional convolution may slide vertically over an input matrix.

Padding:

To handle the edge pixels there are several approaches: Losing the edge pixels Padding with zero value pixels Reflection padding Reflection padding is by far the best approach, where the number of pixels needed for the convolutional kernel to process the edge pixels are added onto the outside copying the pixels from the edge of the image. For a 3x3 kernel, one pixel needs to be added around the outside, for a 7x7 kernel then three pixels would be reflected around the outside. The pixels added around each side is the dimension, halved and rounded down.

Traditionally in many research papers, the edge pixels are just ignored, which loses a small proportion of the data and this gets increasing worse if there are many deep convolutional layers. For this reason, I could not find existing diagrams to easily convey some of the points here without being misleading and confusing stride 1 convolutions with stride 2 convolutions. With padding, the output from a input of width w and height h would be width w and height h (the same as the input with a single input channel), assuming the kernel takes a stride of one pixel at a time.

Strides:

It is common to use a stride two convolution rather than a stride one convolution, where the convolutional kernel strides over 2 pixels at a time, for example our 3x3 kernel would start at position (1,1), then stride to (1,3), then to 1, 5) and so on, halving the size of the output channel/feature map, compared to the convolutional kernel taking strides of one.

With padding, the output from an input of width w, height h and depth 3 would be the ceiling of width w/2, height h/2 and depth 1, as the kernel outputs a single summed output from each stride. Rectified Linear Unit (ReLU): A Rectified Linear Unit is used as a non-linear activation function. A ReLU says if the value is less than zero, round it up to zero.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Normalisation:
Normalisation is the process of subtracting the mean and dividing by the standard deviation. It transforms the range of the data to be between -1 and 1 making the data use the same scale, sometimes called Min-Max scaling.

It is common to normalise the input features by standardising the data, which involves removing the mean and scaling to unit variance. It is often important the input features are centred around zero and have variance in the same order. With some data, such as images the data is scaled so that it's range is between 0 and 1, most simply dividing the pixel values by 255.

Batch normalisation:
Batch normalisation has the benefits of helping to make a network output more stable predictions, reduce overfitting through regularisation and speeds up training by an order of magnitude. Batch normalisation is the process of carrying normalisation within the scope activation layer of the current batch, subtracting the mean of the batch's activations and dividing by the standard deviation of the batches activations.

This is necessary as even after normalizing the input as some activations can be higher, which can cause the subsequent layers to act abnormally and makes the network less stable. Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. Importantly, batch normalization works differently during training and during inference.

During training (i.e. when using fit() or when calling the layer/model with the argument training=True), the layer normalizes its output using the mean and standard deviation of the current batch of inputs. That is to say, for each channel being normalized, the layer returns gamma * (batch - mean(batch)) / sqrt(var(batch) + epsilon) + beta, where: epsilon is small constant (configurable as part of the constructor arguments) gamma is a learned scaling factor (initialized as 1), which can be disabled by passing scale=False to the constructor. beta is a learned offset factor (initialized as 0), which can be disabled by passing center=False to the constructor.

During inference (i.e. when using evaluate() or predict() or when calling the layer/model with the argument training=False (which is the default), the layer normalizes its output using a moving average of the mean and standard deviation of the batches it has seen during training. That is to say, it returns gamma * (batch - self.moving_mean) / sqrt(self.moving_var + epsilon) + beta.

self.moving_mean and self.moving_var are non-trainable variables that are updated each time the layer in called in training mode, as such:
moving_mean = moving_mean * momentum + mean(batch) * (1 - momentum)
moving_var = moving_var * momentum + var(batch) * (1 - momentum)

As such, the layer will only normalize its inputs during inference after having been trained on data that has similarstatistics as the inference data.

Arguments
axis: Integer, the axis that should be normalized (typically the features axis). For instance, after a Conv2D layer with data_format="channels_first", set axis=1 in BatchNormalization.
momentum: Momentum for the moving average.
epsilon: Small float added to variance to avoid dividing by zero. **center**: If True, add offset of beta to normalized tensor. If False, beta is ignored.
scale: If True, multiply by gamma. If False, gamma is not used. When the next layer is linear (also e.g. nn.relu), this can be disabled since the scaling will be done by the next layer. **beta_initializer**: Initializer for the beta weight.
gamma_initializer: Initializer for the gamma weight. **moving_mean_initializer**: Initializer for the moving mean. **moving_variance_initializer**: Initializer for the moving variance.
beta_regularizer: Optional regularizer for the beta weight. **gamma_regularizer**: Optional regularizer for the gamma weight.
beta_constraint: Optional constraint for the beta weight.
gamma_constraint: Optional constraint for the gamma weight.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

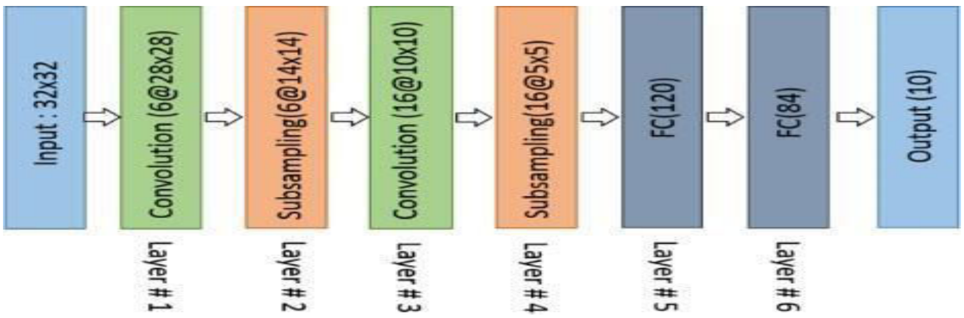
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Call arguments
inputs: Input tensor (of any rank).
training: Python boolean indicating whether the layer should behave in training mode or in inference mode.

LIST OF MODULES
Manual Architecture Lent Architecture Alexnet Architecture MLP, NLTK
Deployment
TYPES OF CNN:
ALEXNET LENET

ALEXNET:
AlexNet architecture consists of 5 convolutional layers, 3 max- pooling layers, 2 normalization layers, 2 fully connected layers, and 1 softmax layer. 2. Each convolutional layer consists of convolutional filters and a nonlinear activation function ReLU. 3. The pooling layers are used to perform max pooling. AlexNet contained eight layers; the first five were convolutional layers, some of them followed by max-pooling layers, and the last three were fully connected layers. It used the non-saturating ReLU activation function, which showed improved training performance over tanh and sigmoid.

Architecture of AlexNet:



Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.

Pooling layers:
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

Dense or Fully connected layers:
Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an MLP.

Convolutional layers:
Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.

Pooling layers:
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

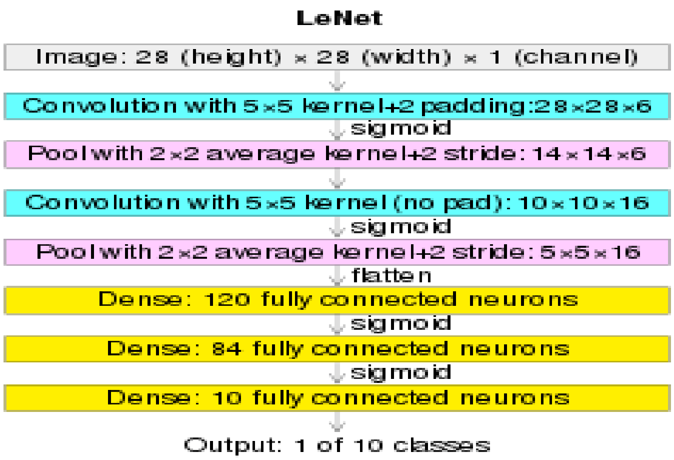
International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Dense or Fully connected layers:
Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an MLP.

LENET
As a representative of the early convolutional neural network, LeNet possesses the basic units of convolutional neural network, such as convolutional layer, pooling layer and full connection layer, laying a foundation for the future development of convolutional neural network. As shown in the figure (input image data with 32*32 pixels) LeNet-5 consists of seven layers. In addition to input, every other layer can train parameters. In the figure, Cx represents convolution layer, Sx represents sub- sampling layer, Fx represents complete connection layer, and x represents layer index.

Architecture of LeNet:



Architecture of LeNet

Convolutional layers:
Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.

Pooling layers:
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

Dense or Fully connected layers:
Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an MLP.
Convolutional layers:
Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.

Pooling layers:
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Dense or Fully connected layers:

Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an MLP.

IV. FUTURE WORK

In this research, we developed an advanced Convolutional Neural Network (CNN)-based model for accurate diagnosis of dog skin diseases. While the current study demonstrates promising results, there are several areas for further improvement and expansion. Future work will focus on the following key aspects. To enhance the robustness of the model, future studies will include a larger and more diverse dataset covering various breeds, age groups, and environmental conditions. Collecting high-quality labeled images from multiple veterinary sources and real-world settings will help improve model generalization.

V.CONCLUSION

In this study, we explored deep learning approaches for accurate dog skin disease diagnosis using advanced Convolutional Neural Networks (CNNs). Skin diseases in dogs can lead to severe health complications if not diagnosed and treated promptly.

Traditional diagnostic methods often rely on clinical observation and laboratory testing, which can be time-consuming and prone to human error. To address these challenges, we implemented a deep learning-based diagnostic system that leverages CNN architectures for automatic detection and classification of various canine skin diseases.Our approach involved training CNN models on a dataset of labeled skin disease images, allowing the network to learn distinguishing features of different conditions. Through rigorous model evaluation, we achieved high accuracy in classification, demonstrating the potential of deep learning in veterinary diagnostics. The system effectively differentiates between normal skin conditions and diseases such as fungal infections, bacterial dermatitis, and parasitic infestations.The experimental results confirm that CNN-based models can significantly enhance diagnostic precision and reduce dependency on manual examination. By integrating this technology into veterinary practices, early detection of skin diseases can be improved, leading to faster treatment and better health outcomes for dogs. Future work can involve expanding the dataset, incorporating additional deep learning techniques such as attention mechanisms, and developing a user-friendly application for veterinarians and pet owners.Overall, this research highlights the effectiveness of deep learning in improving diagnostic accuracy for canine dermatological conditions, paving the way for more efficient and accessible veterinary healthcare solutions.

REFERRENCES

[1] Bagra J., Nair S., Athira V., Kumar M., Kumar M., Thomas P., Kumar B., Chaturvedi V., Dandapat P., and Abhishek., “In Vitro Virulotyping, Antifungal Susceptibility Testing and DNA Fingerprinting of Microsporum Canis Strains of Canine and Feline Origin,” Journal of Comparative Immunology, Microbiology and Infectious Diseases, vol. 104, pp. 102-100, 2024.

[2] Bernicker M., Birrer C., Seeger M., Almeida B., Vogel F., and Cargnelutti J., “Antimicrobial Activity of Cationic Water-Soluble Porphyrin against Multidrug-Resistant Bacteria in Biofilms and Canine Skin Samples,” World Journal of Microbiology and Biotechnology, vol. 40, no. 4, pp. 1 -9, 2024. DOI: 10.1007/s11274-024-03939-7

[3] Calabro C., Sadhu R., Xu Y., Aprea M., Guarino C., and Cazer C., “Longitudinal Antimicrobial Susceptibility Trends of Canine Staphylococcus Pseudintermedius,” Journal of Preventive Veterinary Medicine, vol. 226, pp. 106170, 2024. https://doi.org/10.1016/j.prevetmed.2024.106170

[4] Cugmas B. and Olivry T., “Evaluation of Skin Erythema Severity by Dermatoscopy in Dogs with Atopic Dermatitis,” Journal of Veterinary Dermatology, vol. 32, no. 2, pp. 183-e46, 2021. DOI:10.1111/vde.12932

[5] Emanuelli M., Kommers G., Antoniazzi A., Bernardes F., Lopes S., and Fighera R., “Myoepithelial Cells and Extracellular Matrix in the Cytologic Differentiation of Canine Mammary Tumors,” Journal of Veterinary Clinical Pathology, vol. 49, no. 3, pp. 451-458, 2020. DOI:10.1111/vcp.12894

[6] Harrand R., Feline Reticulocytes: Microscopy Images of Different Cell Types, https://www.kaggle.com/tentotheminus9/feline reticulocytes, Last Visited, 2024.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[7] Jones C. and Murugamani C., "Malaria Parasite Detection on Microscopic Blood Smear Images with Integrated Deep Learning Algorithms," The International Arab Journal of Information Technology, vol. 20, no. 2, pp. 170-179, 2023. doi:10.34028/iajit/20/2/3

[8] Kelly P., McKay J., Maguire D., Jones M., Roberts L., Powell F., and Breathnach R., "A Retrospective Study of Cases of Canine Demodicosis Submitted to a Commercial Diagnostic Laboratory Servicing the United Kingdom and Ireland (2017-2018) Part 2, Aerobic Culture and Antimicrobial Susceptibility Results," Journal of Research in Veterinary Science, vol. 153, pp. 92-98, 2022. doi:10.1016/j.rvsc.2022.10.021.

[9] Marques G., Rocha L., Vargas T., Pulz L., Huete G., Cadrobbi K., Pires C., Sanches D., Mota E., and Strefezzi R., "Relationship of Galectin-3 Expression in Canine Cutaneous Squamous Cell Carcinomas with Histopathological Grading and Proliferation Indices," Journal of Comparative Pathology, vol. 178, pp. 16-21, 2020.DOI:10.1016/j.jcpa.2020.06.004.Munday J., Knight C., and Luff J., "Papillomaviral Skin Diseases of Humans, Dogs, Cats and Horses: A Comparative Review. Part 2: Pre- Neoplastic and Neoplastic Diseases," The Veterinary Journal, vol. 288, pp. 105898, 2022. https://doi.org/10.1016/j.tvjl.2022.105898.

[10] Pardo-Marin L., Ceron J., Tecles F., Baneth G., and Martínez-Subiela S., "Comparison of Acute Phase Proteins in Different Clinical Classification Systems for Canine Leishmaniosis," Journal of Veterinary Immunology and Immunopathology, vol. 219, pp. 109958, 2020. DOI:10.1016/j.vetimm.2019.109958.

[11] Raigonda M. and Shweta., "Signature Verification System Using SSIM in Image Processing," Journal of Scientific Research and Technology, vol. 2, no. 1, pp. https://doi.org/10.61808/jsrt79 5-11,
2024.

[12] Saengchoowong S., Jitvaropas R., Poomipak W., Praianantathavorn K., and Payungporn S., "Identification of Bacteria Associated with Canine Otitis Externa Based on 16S R-DNA High Throughput Sequencing," Brazilian Journal of Microbiology, vol. 54, no. 4, pp. 3283-3290, 2023. DOI: 10.1007/s42770-023-01166-0.

[13] Sözmen M., Devrim A., Sudağıdan M., Kabak Y., and Yıldırım F., "Expression of Angiogenic Growth Factors in Canine Squamous Cell Cancers," Journal of Biotechnic and Histochemistry, vol. 96, no. 6, pp. 450-459, 2021. DOI:10.1080/10520295.2020.1818826.

[14] Starr H., Howerth E., Jr R., Barber J., Leon R., Blubaugh A., and Banovic F., "Characterization of the Serum and Skin Inflammatory Profile in Canine Pemphigus Foliaceus Using Multiplex Assay and Quantitative Real-Time Polymerase Chain Reaction (QRT-PCR)," Veterinary Immunology and Immunopathology Journal, vol. 262,
pp. 110631, 2023..

[15] Stempelová L., Kubašová I., Bujňáková D., Kačírová J., Farbáková J., Maďar M., Karahutová L., and Strompfová V., "Distribution and Characterization of Staphylococci Isolated from Healthy Canine Skin," Topics in Companion Animal Medicine, vol. 49, pp. 100665, 2022. https://doi.org/10.1016/j.tcam.2022.100665.

[16] Taranum A. and Mahesh S., "A Survey Analysis for the Detection of Canine Diseases among Domestic Mammals Using Image Texture Pattern Extraction Methods," European Journal of Molecular and Clinical Medicine, vol. 9, no. 7, pp. 7385-7392, 2022 .

[17] Taranum A. and Mahesh S., "An Optimal Texture Pattern Model of Big Data Processing for Canine Disease Classification," International Journal of Intelligent Systems and Applications Engineering, vol. 12, no. 2, pp. 458-466, 2023.

[18] Upadhyay A., Singh G., Mhatre S., and Nadar P., "Dog Skin Diseases Detection and Identification Using Convolutional Neural Networks," SN Computer Science, vol. 4, no. 250, 2023

[19] Xie T., Lin J., Lin D., Zhang D., Xu X., Zhu N., and Lin J., "In Vitro and in Vivo Antibacterial Studies of Volatile Oil from Atractylodis Rhizoma Against Staphylococcus Pseudintermedius and Multidrug Resistant Staphylococcus Pseudintermedius Strains from Canine Pyoderma," Journal of Ethnopharmacology, vol. 319, pp. 117326, 2024..

[20] Veloso Soares E., Nascimento Gonçalves I., Silveira T., Espirito Santo J., Vieira Figueiredo L., Varaschin M., Dantas Cassali G., Del Puerto H., and Ferreira E., "ZEB and Snail Expression Indicates Epithelial-Mesenchymal Transition in Canine Melanoma," Research in Veterinary Sciences, vol. 131, pp. 7-14, 2020.

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com