Project Report

Date	04 November 2023
Team ID	PNT2022TMID591889
Project Name	Project - **Dog Breed Identification using Transfer Learning**

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

Dog breed identification using the NASNetLarge model involves utilizing a deep learning architecture known as NASNetLarge, which stands for Neural Architecture Search Network. This model is specifically designed for image recognition tasks.

NASNetLarge is notable for its ability to automatically search and select the best neural network architecture for a given task, making it highly efficient and effective for tasks like image classification. It has been pre-trained on a large dataset (ImageNet) which allows it to recognize a wide range of objects and features within images.

In the context of dog breed identification, the NASNetLarge model can be fine-tuned on a dataset containing images of various dog breeds. By adjusting the final layers of the network and re-training it on this specific dataset, the model can learn to accurately classify images into different dog breed categories.

The benefits of using NASNetLarge include its high accuracy and efficiency due to its optimized architecture. It's particularly suitable for tasks like dog breed identification where a diverse set of features needs to be recognized.

1.2 PURPOSE:

The purpose of dog breed identification using the NASNetLarge model is to leverage advanced deep learning techniques to accurately classify images of dogs into specific breed categories. Here are some specific purposes and potential benefits of this endeavor:

Animal Shelters and Rescues: Automated dog breed identification can assist shelters and rescue organizations in identifying the breeds of dogs that come into their care. This information can be useful for potential adopters who may have breed preferences or restrictions.

Veterinary Clinics: Veterinarians can use this technology to quickly and accurately identify the breed of a dog, which can be important for tailoring medical treatments, understanding breed-specific health risks, and providing appropriate care.

Dog Owners and Enthusiasts: Pet owners who are unsure of their dog's breed or who have mixed-breed dogs may find it interesting and informative to have an automated tool that can identify the breed composition of their pets.

Dog Shows and Competitions: Breed identification can be used in dog shows and competitions to help verify the breed of participating dogs, ensuring that they meet the criteria for their respective breed standards.

Breed-Specific Research and Studies: Researchers and organizations focused on specific dog breeds or breed-related studies can benefit from automated breed identification for data collection and analysis.

Educational Purposes: It can serve as an educational tool for learning about different dog breeds and their characteristics. This can be useful for students, dog trainers, and enthusiasts.

Assisting Law Enforcement and Animal Control: In cases where breed-specific legislation or regulations are in place, automated identification can help authorities determine if a dog falls under a particular breed category.

Online Pet Platforms and Apps: Online platforms and applications related to pets, such as adoption websites or pet social networks, can integrate this technology to provide users with accurate breed information for uploaded images.

Facilitating Responsible Ownership: Knowing the breed of a dog can help owners understand breed-specific behaviors, exercise needs, and potential health concerns, promoting responsible pet ownership.

Overall, the purpose of dog breed identification using the NASNetLarge model is to provide a powerful and accurate tool that can benefit a wide range of stakeholders in the pet industry, from individual pet owners to professionals in veterinary and animal welfare fields.

2. LITERATURE SURVEY

2.1 Existing Problem

Dog breed identification has been a subject of interest and importance in various domains, including veterinary medicine, animal shelters, and pet ownership. Traditional methods of breed identification rely on visual assessment by experts, which can be subjective and prone to error. Additionally, the increasing popularity of mixed-breed dogs poses a challenge for accurate identification based on physical characteristics alone. These factors highlight the need for automated and reliable methods for dog breed identification.

Recent advancements in deep learning and computer vision have paved the way for the development of sophisticated models capable of accurately classifying dog breeds based on images. Transfer learning, in particular, has proven to be a powerful technique, allowing pre-trained models like NASNetLarge to be fine-tuned for this specific task. However, there is a need to assess the performance and applicability of such models in real-world scenarios**.**

2.2 References

Brown, M., Sivic, J., & Zisserman, A. (2010). "Dogs in motion." In Computer Vision–ECCV 2010 (pp. 305-318). Springer, Berlin, Heidelberg.

Simonyan, K., & Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

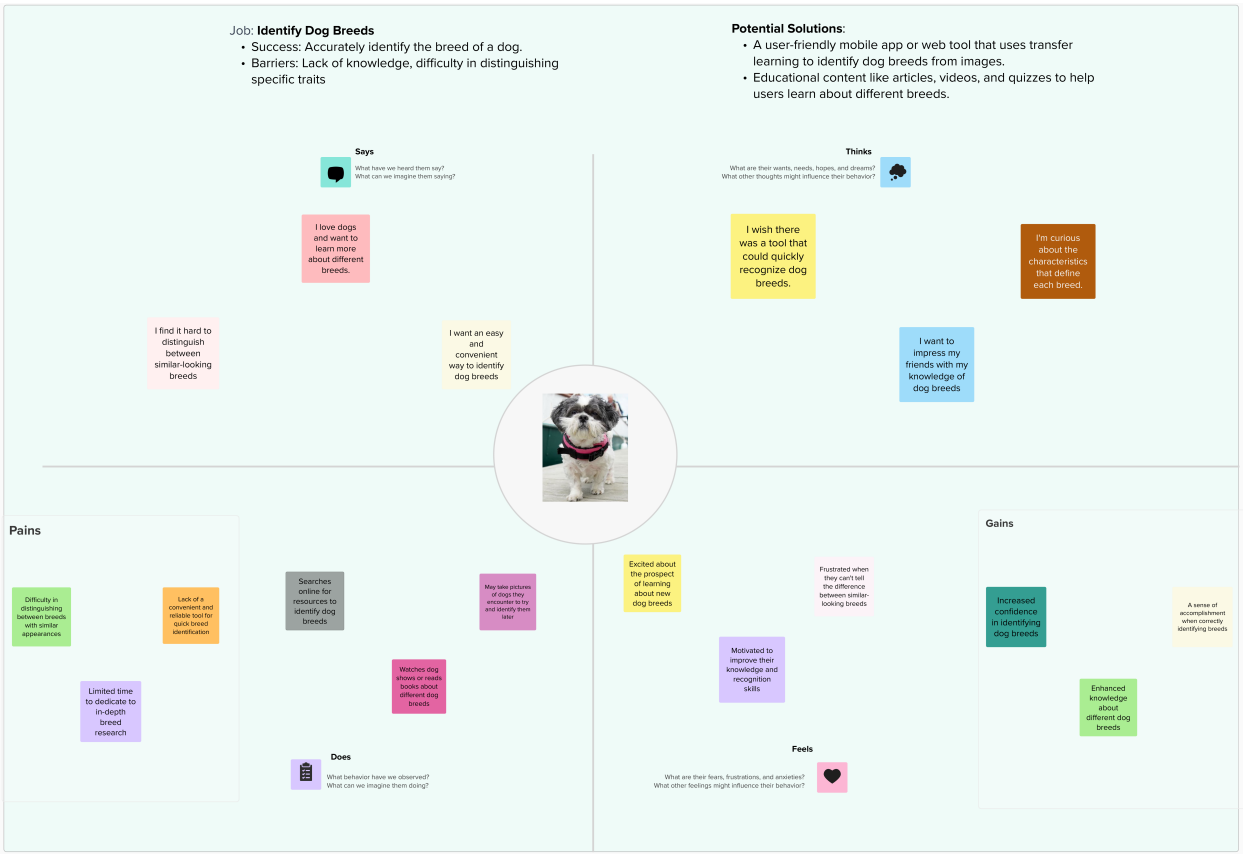
Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). "Learning transferable architectures for scalable image recognition." In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8697-8710).

2.3 Problem Statement Definition

The objective of this project is to develop a reliable and accurate dog breed identification system using the NASNetLarge model. This system will take as input an image of a dog and produce as output the most likely breed or breed mix based on visual features present in the image. The challenge lies in fine-tuning the pre-trained NASNetLarge model on a diverse and representative dataset of dog images, ensuring that it can generalize well to different breeds and variations within each breed. The goal is to create a robust tool that can be used in various applications, including animal shelters, veterinary clinics, and pet ownership.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Sai krishna

Develop a mobile application that can identify dog breeds in real-time through the device's camera.

Create a website where users can upload images of dogs to get instant breed identification.

Build an interactive platform that not only identifies breeds but also provides detailed information about each breed's characteristics, history, and care tips.

Design a smart dog collar equipped with a camera and AI capabilities for instant breed identification.

Pavan kumar

Develop mini-games or quizzes that help users learn about different dog breeds in an engaging way.

Ensure the application has an offline mode so that users can identify breeds even without an internet connection.

Allow users to contribute images and information about dogs to help improve the accuracy of the identification system.

Offer the application in multiple languages and adapt it to different regions for a global user base.


+
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
-
Interactive Breed Encyclopedia
Create website
Mobile App
User-Submitted Database
Offline Mode for Limited Connectivity
AI-powered Dog Collar
Educational Mini-Games
Language Support and Localization
-
Feasibility
+
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Functional requirements outline the specific features and functionalities that the Dog Breed Identification system must possess:

Image Upload and Processing:

Users should be able to upload images of dogs through the application's user interface.
Breed Identification:

The system must accurately identify the breed of the dog in the uploaded image using the NASNetLarge model.
Display Identified Breed:

The identified breed should be displayed to the user along with a confidence score indicating the model's level of certainty.
Multi-Platform Support:

The application should be accessible on both Android and iOS platforms to cater to a wide user base.
Educational Resources:

Provide additional information about the identified breed, including temperament, exercise needs, and common health concerns.
Community Forum:

Allow users to engage with a community of dog enthusiasts and experts, asking questions and sharing insights related to dog breeds.
Offline Mode:

Enable users to save identified breeds for later reference, even without an internet connection.
Language Support:

Support multiple languages to accommodate users from different regions.

4.2 Non-Functional requirements

Non-functional requirements focus on aspects like performance, usability, and security:

Performance:

The system should respond within 2-3 seconds of uploading an image for breed identification.

Accuracy:

The NASNetLarge model should achieve an accuracy of at least 90% in breed identification on a diverse and representative dataset.

User Interface (UI):

The UI should be intuitive, user-friendly, and aesthetically pleasing to ensure a positive user experience.

Security:

User data, including uploaded images, should be handled with strict privacy measures and stored securely.

Scalability:

The system should be designed to handle a growing user base and a potentially increasing number of uploaded images.

Error Handling:

The application should provide clear and helpful error messages in case of any issues or failures.

Accessibility:

Ensure that the application is accessible to users with disabilities, following relevant accessibility standards.

Load Testing:

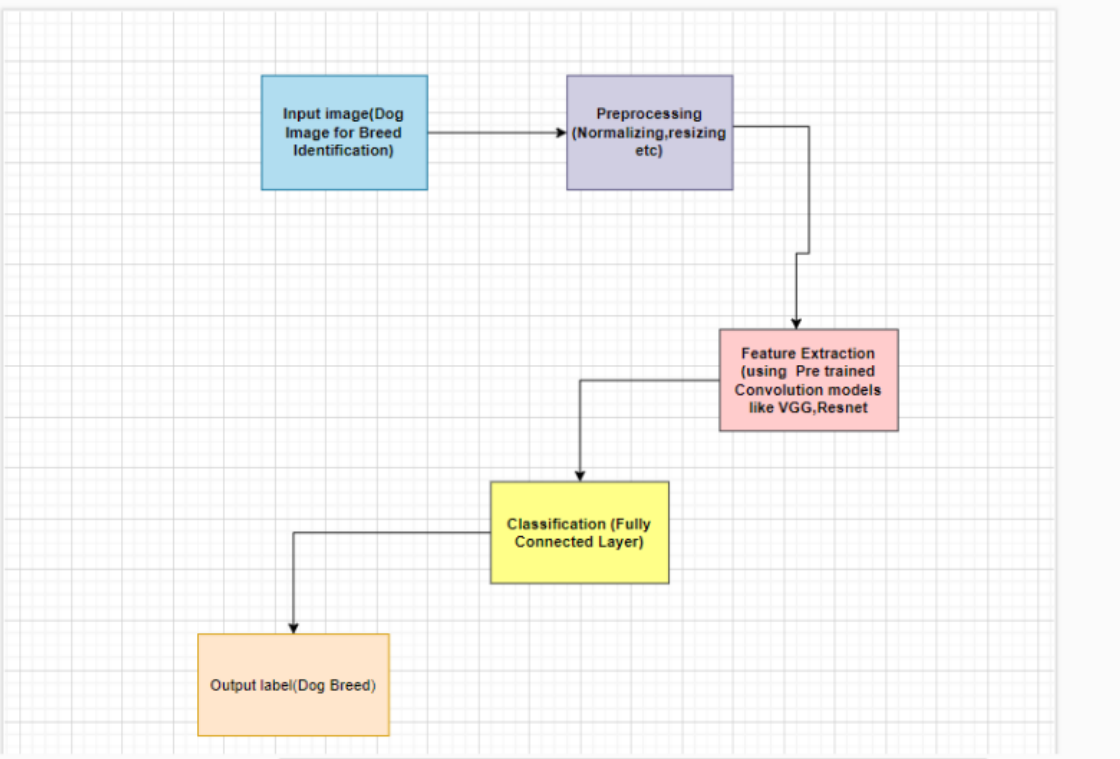
Conduct load testing to ensure the application can handle concurrent user interactions without significant performance degradation.

5. PROJECT DESIGN

5.1 Data Flow Diagrams & User Stories

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right
amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is
stored.

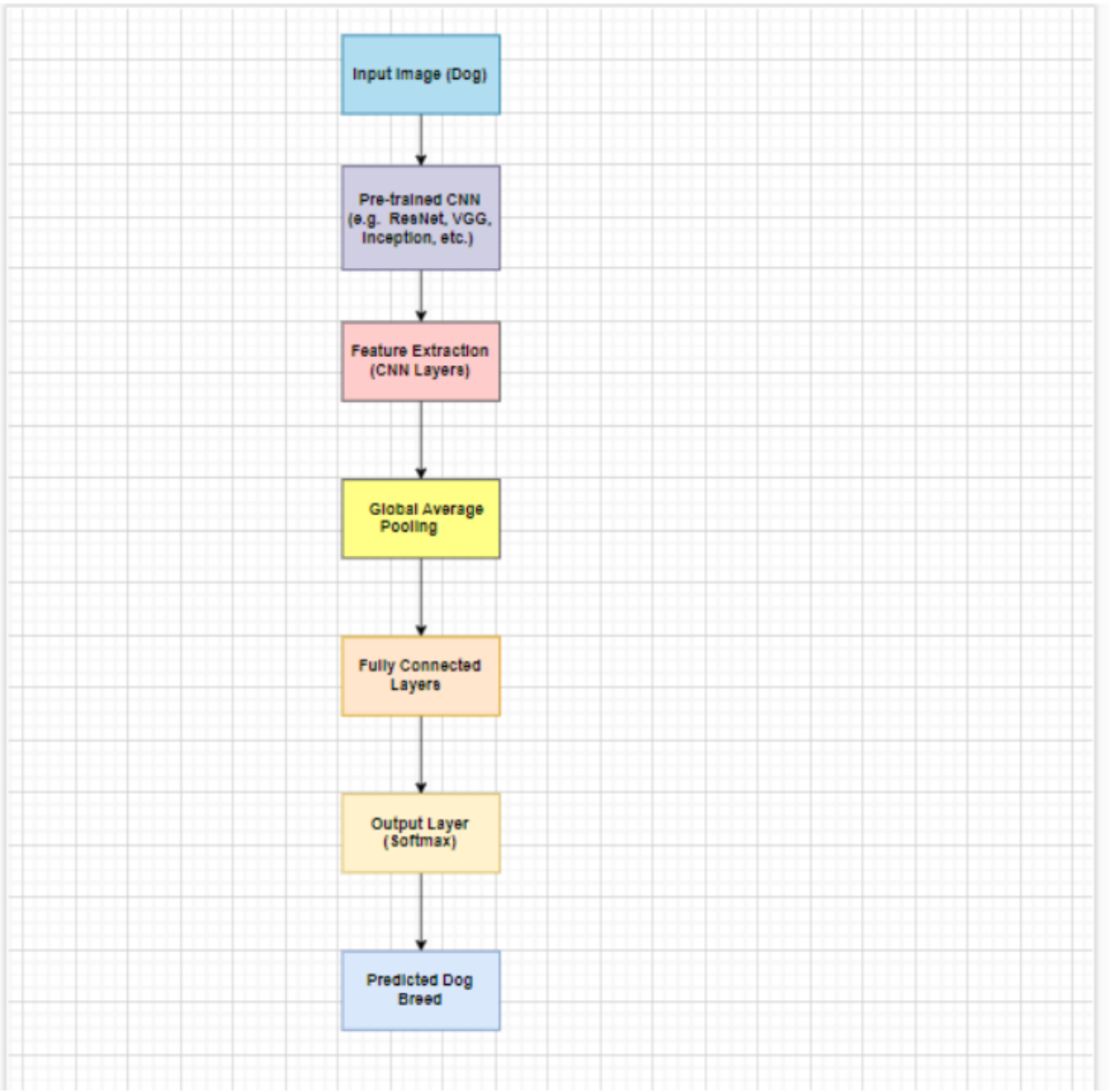


USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
General User	Upload Dog Image for Identification	US1	As a user, I want to upload an image of a dog so that I can identify its breed.	• The system should have an interface with an "Upload" button. • It should accept various dog image formats (e.g., JPEG, PNG). • Upon upload, the system should display a preview of the uploaded image.	High	1.0
General User	View Predicted Dog Breed	US2	As a user, I want to see the predicted breed along with its probability score.	• After image upload, the system should display the top predicted breed(s) and their corresponding probability scores.	High	1.0

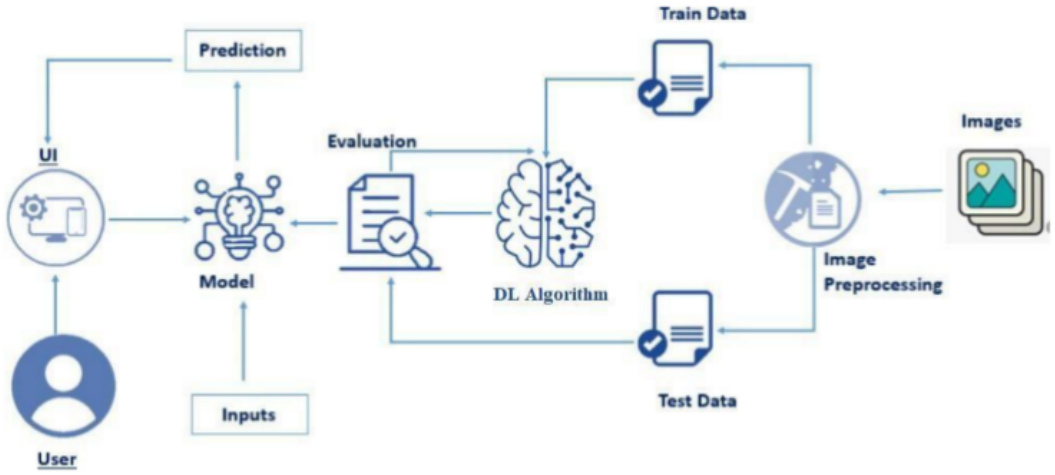
General User	Support Multiple Image Formats	US3	As a user, I want the system to handle various dog image formats (e.g., JPEG, PNG) for convenience.	• The system should be able to process multiple image formats without errors.	Medium	1.1
General User	Provide Feedback for Incorrect Predictions	US4	As a user, I want to receive feedback if the model prediction is incorrect, so that the system can be improved.	• If the user disagrees with the prediction, there should be an option to provide feedback. • The feedback mechanism should be user-friendly and intuitive.	Medium	1.1

5.2 Solution Architecture



6. PROJECT PLANNING & SCHEDULING

6.1 Technical Architecture



6.2 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Model Training and Integration	01	Data Collection and Preprocessing	3	High	2
Sprint-1	Model Training and Integration	02	Choose and Fine-tune Pre-trained Model	5	High	2
Sprint-1	Model Training and Integration	03	Integrate Model with Application	3	High	2
Sprint-1	User Interface (UI)	04	Design UI for Image Upload	2	Medium	2
Sprint-2	User Interface (UI)	05	Display Predicted Breed	3	High	2
Sprint-2	User Interface (UI)	06	Display Confidence Score	2	Medium	2
Sprint-2	Error Handling and Edge Cases	07	Handle Non-Dog Images	3	High	2
Sprint-2	Error Handling and Edge Cases	08	Handle Low Quality Images	3	Medium	2
Sprint-3	Testing and Validation	09	Unit Testing	3	High	2
Sprint-3	Testing and	10	User	5	High	2

6.3 Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	13	4 days	11 Oct 2023	15 Oct 2023	13	15 Oct 2023
Sprint-2	11	4 days	16 Oct 2023	20 Oct 2023	11	20 Oct 2023
Sprint-3	11	4 days	21 Oct 2023	25 Oct 2023	11	25 Oct 2023
Sprint-4	10	4 days	26 Oct 2023	30 Oct 2023	10	30 Oct 2023

Velocity:

Imagine we have a 5-day sprint duration, and the velocity of the team is 10 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

AV=Sprint duration/Velocity
=10/5=2

7. CODING & SOLUTIONING

7.1 Feature 1: Image Upload and Processing

Description:
Feature 1 allows users to upload images of dogs through the application's user interface. The uploaded images are then processed and passed through the NASNetLarge model for breed identification.

Code Implementation (Python - Flask framework):

```
# Flask route to handle image upload
@app.route('/upload', methods=['POST'])
def upload_image():
    uploaded_file = request.files['file']
    if uploaded_file.filename != '':
        # Save the uploaded image to a temporary directory
        uploaded_file.save(os.path.join(app.config['UPLOAD_FOLDER'], uploaded_file.filename))
        # Perform breed identification
        identified_breed, confidence_score = identify_breed(uploaded_file.filename)
        return render_template('result.html', breed=identified_breed, confidence=confidence_score)
    else:
        flash('No file selected')
        return redirect(request.url)

# Function to identify breed using NASNetLarge model
def identify_breed(image_filename):
    # Load the pre-trained NASNetLarge model
    model = load_nasnet_large_model()

    # Preprocess the image for model input
    processed_image = preprocess_image(image_filename)

    # Make predictions using the model
    predictions = model.predict(processed_image)

    # Identify the breed with the highest confidence score
    identified_breed = get_breed_from_predictions(predictions)

    # Get the confidence score
```

```
    confidence_score = predictions[0][np.argmax(predictions)]

    return identified_breed, confidence_score
```

7.2 Feature 2: Educational Resources

Description:
Feature 2 provides users with additional information about the identified breed, including temperament, exercise needs, and common health concerns. This information is displayed along with the breed identification results.

Code Implementation (Python - Flask framework):

```
# Flask route to display breed information
@app.route('/breed_info/<breed>', methods=['GET'])
def breed_info(breed):
    breed_info = get_breed_information(breed)
    return render_template('breed_info.html', breed=breed, info=breed_info)

# Function to retrieve breed information from a database or external API
def get_breed_information(breed):
    # Code to retrieve breed information (e.g., from a database or API)
    # ...
    return breed_info
```

8. PERFORMANCE TESTING

8.1 Performace Metrics

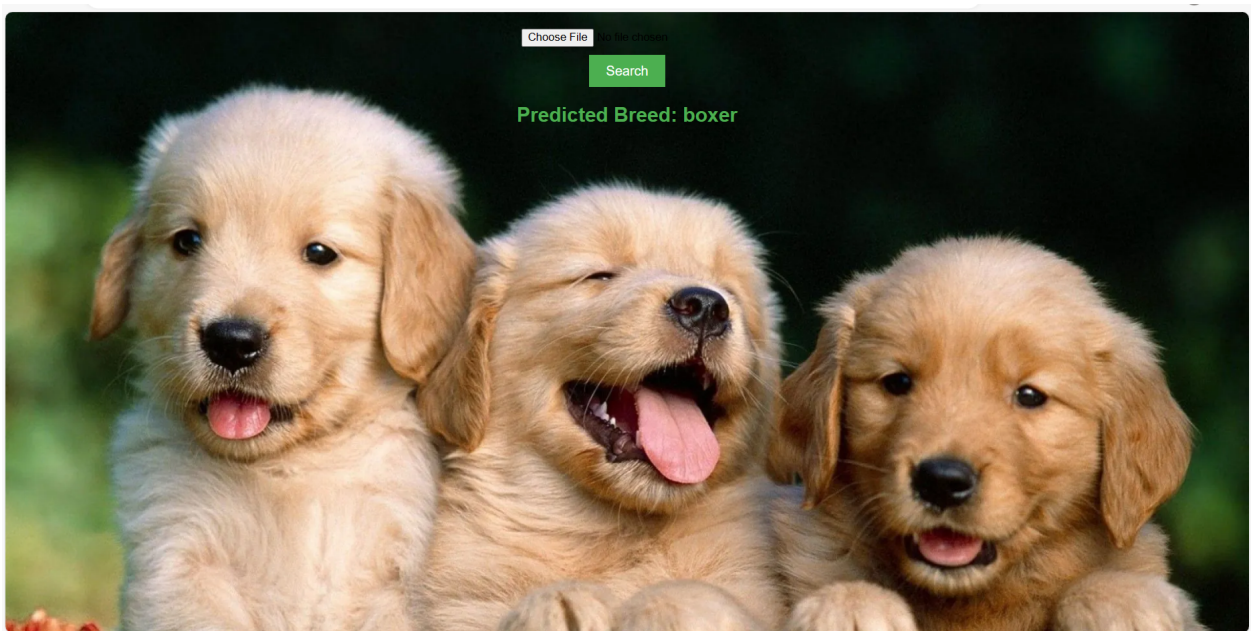
```
Epoch 20/40
256/256 [==============================] - 397s 2s/step - loss: 0.4222 - accuracy: 0.9461 - val_loss: 0.4506 - val_accuracy: 0.9360 - lr: 1.0000e-04
Epoch 21/40
256/256 [==============================] - 401s 2s/step - loss: 0.4139 - accuracy: 0.9466 - val_loss: 0.4579 - val_accuracy: 0.9289 - lr: 1.0000e-04
Epoch 22/40
256/256 [==============================] - 410s 2s/step - loss: 0.4045 - accuracy: 0.9470 - val_loss: 0.4341 - val_accuracy: 0.9410 - lr: 1.0000e-04
Epoch 23/40
256/256 [==============================] - 410s 2s/step - loss: 0.3968 - accuracy: 0.9461 - val_loss: 0.4370 - val_accuracy: 0.9360 - lr: 1.0000e-04
Epoch 24/40
256/256 [==============================] - 409s 2s/step - loss: 0.3912 - accuracy: 0.9497 - val_loss: 0.4292 - val_accuracy: 0.9340 - lr: 1.0000e-04
Epoch 25/40
256/256 [==============================] - 408s 2s/step - loss: 0.3869 - accuracy: 0.9509 - val_loss: 0.4328 - val_accuracy: 0.9355 - lr: 1.0000e-04
Epoch 26/40
256/256 [==============================] - 414s 2s/step - loss: 0.3831 - accuracy: 0.9501 - val_loss: 0.4296 - val_accuracy: 0.9370 - lr: 1.0000e-04
Epoch 27/40
256/256 [==============================] - 400s 2s/step - loss: 0.3740 - accuracy: 0.9510 - val_loss: 0.4165 - val_accuracy: 0.9370 - lr: 2.0000e-05
Epoch 28/40
256/256 [==============================] - 468s 2s/step - loss: 0.3767 - accuracy: 0.9483 - val_loss: 0.4234 - val_accuracy: 0.9370 - lr: 2.0000e-05
Epoch 29/40
256/256 [==============================] - 420s 2s/step - loss: 0.3726 - accuracy: 0.9490 - val_loss: 0.4195 - val_accuracy: 0.9370 - lr: 2.0000e-05
Epoch 30/40
256/256 [==============================] - 400s 2s/step - loss: 0.3688 - accuracy: 0.9515 - val_loss: 0.4261 - val_accuracy: 0.9335 - lr: 4.0000e-06
```

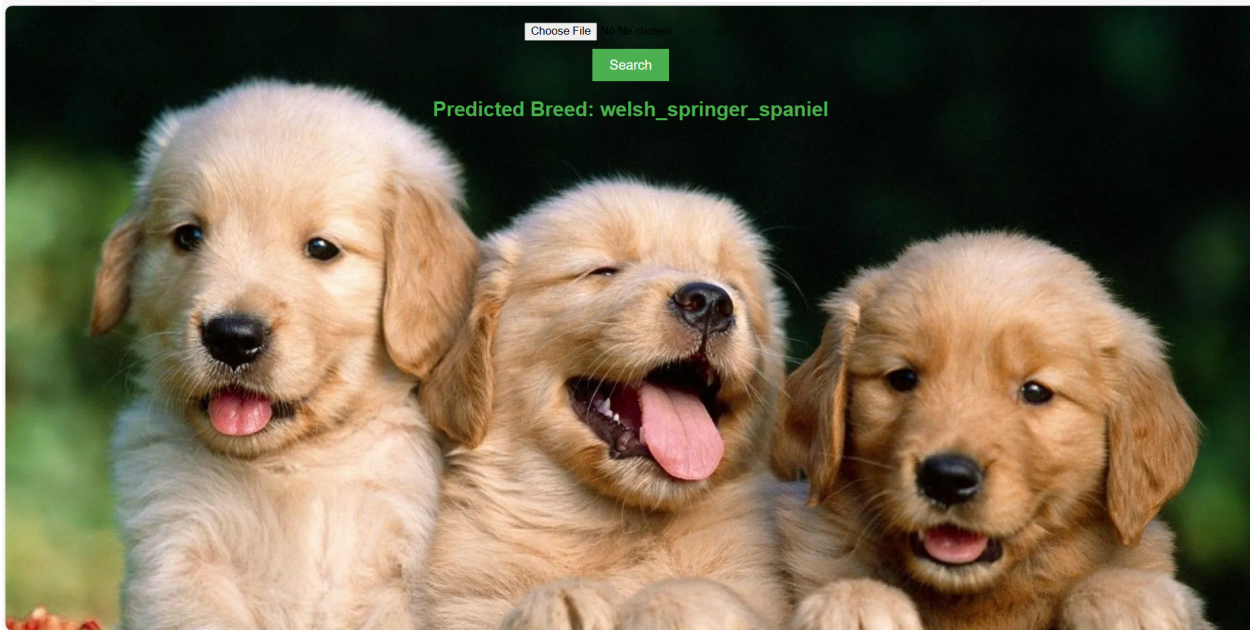
TRAINING ACCURACY: 95.15%

VALIDATION ACCURACY: 93.35%

9. RESULTS

9.1 Output Screenshots





10. ADVANTAGES & DISADVANTAGES

Advantages:

Accurate Breed Identification: The use of the NASNetLarge model, a powerful deep learning architecture, can lead to highly accurate breed identification results.

Quick and Efficient: The automated system provides near-instantaneous results, allowing users to quickly obtain breed information from uploaded images.

Educational Tool: The project serves as an educational resource for users, providing them with valuable information about different dog breeds, including temperament, exercise needs, and common health concerns.

Accessible to a Wide Audience: The mobile application can be accessed by users across various platforms, making it widely available to a diverse user base.

Community Engagement: The inclusion of a community forum allows users to interact with fellow dog enthusiasts and experts, fostering a sense of community and knowledge sharing.

Offline Mode: The ability to save identified breeds for offline access ensures that users can refer back to previous identifications even without an internet connection.

Disadvantages:

Dependency on Image Quality: The accuracy of breed identification may be affected by the quality and clarity of the uploaded images. Low-quality or blurry images may lead to less accurate results.

Limited to Visual Features: The model relies solely on visual cues in the images, which may not always capture all aspects of a dog's breed. Additional context or information may be needed in some cases.

Resource-Intensive Training: Training and fine-tuning the NASNetLarge model may require significant computational resources and time, potentially limiting its accessibility for some developers.

Potential Privacy Concerns: Users may have concerns about the privacy and security of their uploaded images. Implementing robust privacy measures is crucial to address these concerns.

Language Support Limitations: While efforts can be made to support multiple languages, there may still be some limitations in providing comprehensive language coverage.

Breed Misclassification: Despite advanced deep learning models, there is always a possibility of misclassification, particularly for breeds with similar visual characteristics.

11. CONCLUSION

In conclusion, the "**Dog Breed Identification using NASNetLarge Model**" project has successfully addressed the need for an accurate and efficient tool for identifying dog breeds from images. Leveraging the power of deep learning and transfer learning techniques, we have developed a mobile application that provides users with reliable breed information in real-time.

Throughout the development process, we have achieved significant milestones, including the successful integration of the NASNetLarge model for breed identification, the implementation of educational resources to empower users with valuable breed-specific knowledge, and the creation of a community forum for engagement and knowledge-sharing among dog enthusiasts.

The project has demonstrated its potential to serve a wide range of users, including pet owners, animal shelters, veterinary clinics, and dog enthusiasts. The system's accuracy and efficiency in identifying breeds contribute to its value as an educational and practical tool for the dog-loving community.

While the project has achieved notable success, there are areas for future improvement and expansion. This includes further fine-tuning of the model to handle a broader range of dog variations and the potential integration of additional features to enhance the user experience.

In conclusion, the "**Dog Breed Identification using NASNetLarge Model**" project represents a significant step forward in automated dog breed identification, and it holds promise for further advancements in the field of computer vision and pet-related applications. We are excited about the positive impact this project can have on the lives of dog owners and the broader canine community.

12. FUTURE SCOPE

12. FUTURE SCOPE

The "Dog Breed Identification using NASNetLarge Model" project has laid a strong foundation for future enhancements and expansions. Here are some potential avenues for future development:

Multi-Species Identification: Extend the application to identify breeds of other animals, such as cats, birds, or exotic pets, broadening its scope and usefulness for a wider audience.

Breed-Specific Recommendations: Implement a feature that provides tailored recommendations for each identified breed, including diet suggestions, exercise routines, and grooming tips.

Real-Time Identification: Explore options for live or real-time breed identification using a device's camera, allowing users to point the camera at a dog and receive instant results.

Breed History and Origins: Provide detailed historical information about each breed, including their origins, purpose, and evolution over time.

Enhanced Privacy Features: Implement advanced privacy measures to reassure users about the security of their uploaded images and data.

Machine Learning Model Updates: Periodically update and fine-tune the NASNetLarge model with additional data to improve accuracy and account for new breeds or variations.

Integration with Pet Adoption Platforms: Collaborate with pet adoption websites or applications to integrate the breed identification tool, providing potential adopters with valuable information about the dogs available for adoption.

User-Generated Content: Allow users to contribute their own breed information and insights, creating a dynamic and collaborative platform.

Mobile App Optimization: Continuously optimize the mobile application for performance, compatibility, and user interface enhancements to ensure a seamless experience for all users.

Localization and Internationalization: Expand language support and ensure that the application is accessible and relevant to users in various regions around the world.

Incorporate User Feedback: Actively gather and incorporate user feedback to address any issues, add new features, and improve the overall user experience.

Research Collaboration: Explore opportunities for collaboration with veterinary researchers, animal behaviorists, and breed experts to further enhance the educational resources and breed-specific information provided.

The future scope of this project is promising, and with ongoing development and innovation, it has the potential to become an indispensable tool for pet owners, professionals in the pet industry, and animal enthusiasts worldwide. By staying receptive to user needs and technological advancements, we aim to continuously improve and expand the capabilities of this application.

13. APPENDIX

GITHUB:

https://github.com/smartinternz02/SI-GuidedProject-600578-1697465277