GOLDEN OWL SOLUTION

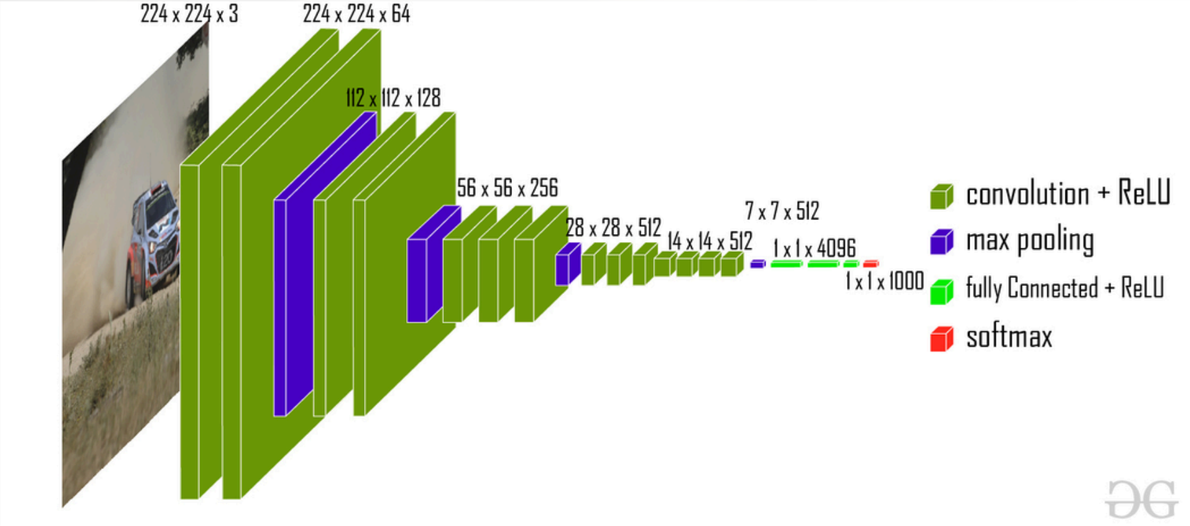
REPORT

Code test 1: Vision AI intern assignment

TASK 1

Image Classification (Cats & Dogs)

MODEL INTRODUCTION
VGG 16



VGG 16 Architecture

Start with a Pre-Trained Model

- VGG16: A deep 16-layer CNN, expert at general image feature extraction from ImageNet.

Adapt for Our Task (Transfer Learning)

- Remove the original 1000-class ImageNet classifier.
- Add a new, custom 2-class (Cat/Dog) classifier.

Train on Custom Data

- Freeze the base layers to keep the learned features.
- Train only the new classifier on the **Cat and Dog** dataset.

DATASET
CAT AND DOG

Dataset Link:
https://www.kaggle.com/datasets/tongpython/cat-and-dog

The Dataset consist of: **10 028** images
Train:

- Cats: **4001** images
- Dogs: **4005** images

Test:

- Cats: **1011** images
- Dogs: **1012** images

Dog image taken from the dataset

Cat image taken from the dataset

PREPROCESS

NORMALIZATION

- All training and validation images were normalized.
- Pixel values were rescaled from the [0, 255] range to the [0, 1] range.

AGUMENTATION

- To reduce overfitting I augmented the dataset
- These included random rotations, zooms, shifts, shear and horizontal flips.

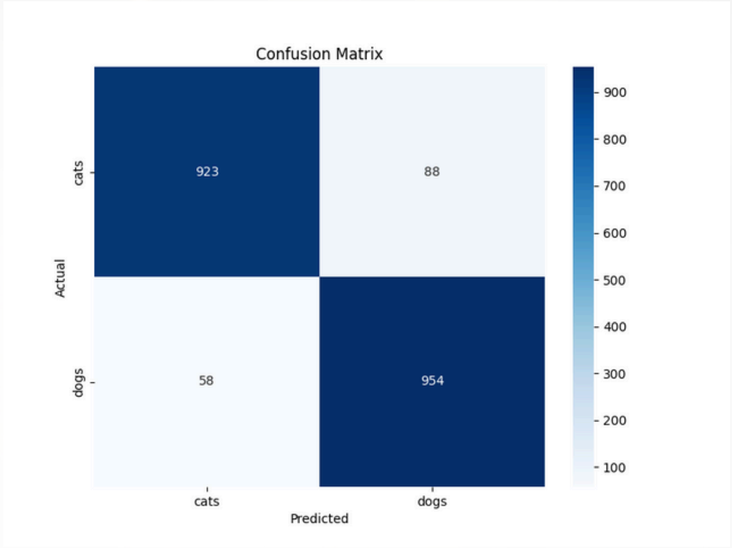
TRAINING RESULT

Training Config:

- Image Size: 224x224
- Batch Size: 32
- Epochs: 25
- Learning Rate: 0.0001
- Loss: Binary Cross-Entropy

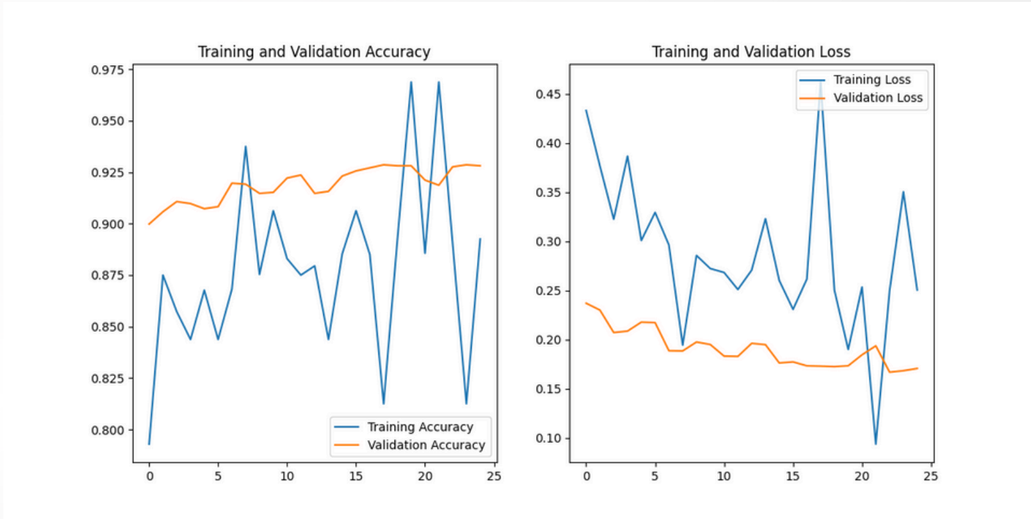
Overall Results:

- Final Training Loss (Epoch 25): 0.2549
- Final Training Accuracy(Epoch 25): 89.0%
- Final Validation Loss (Epoch 25): 0.1705
- Final Validation Accuracy(Epoch 25): 92.8%
- Best Validation Loss : 0.1705
- Best Validation Accuracy: 92.9%



Confusion Matrix

Training History



ANALYSIS AND OBSERVATIONS

1. **Excellent Performance**: The model achieved a high peak validation accuracy of approximately **92.9%**, demonstrating its strong predictive power on unseen data.
2. **Strong Generalization & No Overfitting**: The validation loss consistently decreased throughout the training process, and validation accuracy remained higher than training accuracy. This is a clear indicator that the data augmentation and dropout were highly effective at preventing the model from "memorizing" the training data.
3. **Stable Convergence**: The model's performance on the validation set began to plateau around epoch 18, suggesting that the training had reached a point of optimal performance within the 25-epoch cycle.

Conclusion

- The training was a success. The final model is well-trained, robust, and generalizes effectively to new data, achieving a validation accuracy of approximately 92.8%. The use of a pre-trained VGG16 base, combined with data augmentation, proved to be an effective strategy for preventing overfitting and achieving high performance.

FUTURE WORK AND NEXT STEPS

To build upon the successful results of this project, we propose the following next steps:

Implement a Feature-Focused Approach:

1. Instead of classifying the entire image, we can first use an object detection model to locate and crop the image to the face of the cat or dog. This would force the VGG model to train on the most discriminative features and reduce the impact of distracting backgrounds, potentially leading to higher accuracy.
2. Explore Modern Architectures:
3. While VGG16 provided a strong baseline, performance can be enhanced by leveraging more contemporary models.

We can experiment with:

1. Deeper Architectures like ResNet or VGG19: To potentially increase classification accuracy.
2. Efficient Architectures like EfficientNet or MobileNetV2: To achieve faster inference speeds and lower computational costs, which is critical for real-world deployment, without sacrificing accuracy.

TASK 2

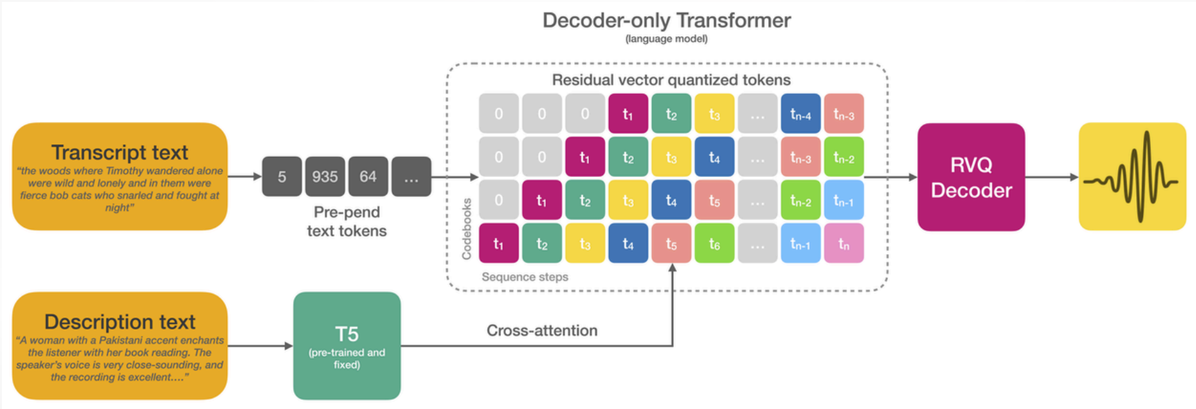
Text to speech

MODEL INTRODUCTION
PARLER

Parler-TTS is not a standard Text-to-Speech (TTS) model; it is a powerful prompt-guided generative audio model developed by Hugging Face. Unlike traditional TTS systems that simply convert text to a specific, pre-determined voice, Parler-TTS operates more like a "digital actor."
Its key innovation is the ability to take two distinct inputs:

- The Content (text): The script of what needs to be said.
- The Description (prompt): A natural language description of how the content should be said. This prompt acts as stage direction, guiding the model's performance.

By conditioning the audio generation on a descriptive prompt, Parler-TTS can produce speech with specific emotions, pitches, speaking rates, and other acoustic properties on the fly, offering a level of creative control that traditional models lack.



Parler Architecture

KEY ADVANTAGES

1

Prompt-Based Style Control: Guide the speech output—including emotion, pitch, and pace—using simple English descriptions instead of complex code.

2

Flexible Style Application: Apply new styles (e.g., "whisper") to any voice on the fly, even without direct training on that specific combination (zero-shot capability).

3

Multi-Speaker Capability: Embed multiple distinct speaker voices into a single model, which can be selected and used through the text prompt.

WHY IT IS USED?

- At first, fine-tuning an English-based model for a tonal language like Vietnamese seems challenging.
- In Vietnamese, the tone of a syllable changes its entire meaning (e.g., ma, má, mà). The key to why this is possible lies in phonetic abstraction.

- **Uses Phonetic Conversion:** The model doesn't learn written Vietnamese; it learns a universal phonetic script.

1

- **G2P Tool is Key**: We use a Grapheme-to-Phoneme (G2P) tool to translate text into these universal sounds.

2

- **Tones are Encoded:** Crucially, this phonetic script explicitly includes the complex Vietnamese tones (e.g., as numbers like AW2).

3

- **Simplified Task:** This transforms the problem from learning a new language to simply learning a new set of sounds.

4

DATASET GOAL: SINGLE-SPEAKER VS. MULTI-SPEAKER

OPTION A: SINGLE-SPEAKER DATASET

- **Goal:** To create a high-fidelity clone of a single, specific voice.
- **Data:**
  - **Audio:** Speaker speech
  - **Transcription:** Subtitle of the speech
  - **Description:** Describe the unique characteristics of that one speaker

OPTION B: MULTI-SPEAKER DATASET

- **Goal:** To create a single model that can generate speech in several different voices.
- **Data:**
  - **Audio:** Speaker speech
  - **Transcription:** Subtitle of the speech
  - **Description:** Describe the unique characteristics of a specific speaker

VOICE DATA GATHERING

I think that we need around ~50-100h of voice data to achieve the optimal results. The following datasets are some sample of multi-speaker (we can filter them out based on the task):

- VLSP 2020: 100h
- FPT Open Speech Dataset (FOSD): 100h
- VietSpeech: 1000h

DataSpeech: Link

DATASET PREPARATION

My objective is to create a robust, in-house data processing pipeline that replicates the functionality of official tools (like Hugging Face dataspeech which is the official repo that create data used by Parler) but is specifically tailored for the Vietnamese language. The pipeline is divided into three main phases: foundational audio processing, multi-stage label generation, and final assembly.

Phase 1: Audio Processing

First, all raw audio is standardized to ensure consistency. This involves resampling to 22,050 Hz, normalizing volume, and trimming silence.

Github Repo: DataSpeech

DATASET PREPARATION

Phase 2: Multi-stage Label Generation
Next, we generate the two required labels for each audio clip:

- A Vietnamese ASR model creates the initial transcript.
- This transcript is then normalized (e.g., numbers to words) and converted into a phonetic script using a G2P (like vi-g2p) tool to create the content text label, ensuring tones are captured.
- Simultaneously, a descriptive English prompt is generated from pre-written templates based on the speaker's identity to create the style prompt label.

Phase 3: Final Assembly

- Finally, the processed audio paths, phonetic text labels, and descriptive prompt labels are combined into a final dataset manifest, ready for training.

Github Repo: DataSpeech

FINE-TUNING PARLER-TTS

Our Approach: We will use the official repo of Parler-TTS to fine-tune the model in Vietnamese.

Reference Link: Parler-TTS

TEXT ENCODER (THE “READER”)

- **Role:** Reads the phonetic script and creates a "blueprint" of the sounds to be made.
- **Challenge:** The pre-trained model only understands English phonemes.
- **Our Goal & The Dataset's Role:** Teach it to read the new Vietnamese phonetic script. Our dataset provides the necessary **phonetic transcripts** (text) and **style prompts** (prompt) for this learning.

AUDIO ENCODER (THE “SPEAKER”)

- **Role:** Takes the "blueprint" from the encoder and generates the final audio waveform.
- **Challenge:** The pre-trained model only knows how to produce English sounds.
- **Our Goal & The Dataset's Role:** Teach it to pronounce the new Vietnamese sounds. The decoder learns this by associating the **phonetic** blueprint with the target audio recordings provided in our dataset.

THANK YOU