# GRU-Based Model – Training Procedure (Your Explanation Style)

To begin with, I implemented a phoneme-level lipreading model that combines a **convolutional encoder** and a **bi-directional GRU**. The overall training setup was based on the aligned `.pt` tensor files I had cached in advance, which made the process much faster.

The architecture itself is structured in two parts:

- A lightweight CNN to extract spatial features from the mouth region in each frame.
- A bi-directional GRU to model the temporal dynamics across time.

In detail:

- Each input sequence consisted of **frames shaped as tensors (T, C, H, W)**.
- The CNN processed every frame individually and projected them into a lower-dimensional embedding space (128D).
- These embeddings were reshaped and passed into the GRU, which processed the temporal sequence.
- The GRU's output was then passed through a fully connected layer to predict phoneme class probabilities for each time step.

I trained the model using the **CTC loss**, which allowed the network to learn without needing one-to-one alignment between frames and phonemes.

The training loop handled:

- Resuming from intermediate checkpoints every 500 batches.
- Batch-level validation to skip problematic data samples.
- Saving the model at the end of every epoch.

I used the Adam optimizer with a learning rate of `1e-4`, batch size of 4, and trained for 5 full epochs on a ~10,000 sample dataset. The entire training was done on CPU, which made it quite slow — but the model was stable throughout.

# ViT-Based Model – Training Procedure

After finishing training the GRU-based model, I implemented a second architecture — this time using a **Transformer-style model** inspired by the **Vision Transformer (ViT)** concept, but adapted for lipreading.

Like the GRU model, this one begins with a CNN to encode each frame. I reused the same convolutional encoder:

- Two convolutional layers followed by ReLU and BatchNorm
- An adaptive average pooling layer that reduces spatial dimensions

Each frame is encoded into a fixed 128-dimensional vector. Once all the frames in a video sequence are encoded, I assemble them into a sequence: one vector per frame.

At that point, instead of feeding the vectors into a GRU, I pass them through a **Transformer encoder**. But since Transformers don't have a built-in sense of order, I first apply a **positional encoding module** to inject temporal information into the sequence.

The Transformer itself is made up of:

- 2 encoder layers
- 4 attention heads per layer
- Feedforward networks of size 256

The output from the Transformer still has the shape `(T, B, D)`, and each time step is passed through a fully connected layer (the classifier) to produce logits over the phoneme vocabulary.

As with the GRU model, I trained using **CTC loss**, and used the same training configuration:

- Batch size = 4
- Learning rate = 1e-4
- Optimizer = Adam
- Device = CPU
- Training duration = 5 epochs on ~10k samples

The training script reused the same caching, collate functions, and checkpointing logic, so the only thing that changed between the two setups was the model architecture.

Despite being relatively lightweight, the ViT-based model performed significantly better — especially in how it captured subtle frame-to-frame transitions. This confirms the benefit of self-attention for visual sequence modeling.

## Benchmarking GRU vs. ViT Models

After training both models for 5 epochs each on a ~10,000-sample subset of LRS2, I evaluated them on a **separate validation set** made of short, unseen video clips. Both models were trained using the same dataset loader, CTC loss, and optimizer configuration — which ensured a fair comparison.

| Model | Phoneme Error Rate (PER) | Params | Notes |
|---|---|---|---|
| **GRU** | 20.35% | 294k | Sequential, baseline model |
| **ViT-style** | 3.22% | 351k | Self-attention, positional encoding |

**Interpretation:**

- The GRU model performed reasonably but struggled to model lip motion across time as effectively.
- The ViT-style model captured temporal dynamics much better, even with only 2 Transformer layers.
- Despite having a similar number of parameters, the ViT architecture achieved significantly lower error, suggesting stronger sequence modeling capabilities.

I also confirmed that both models were evaluated fairly:

- Validation sequences were unseen during training.
- Data was properly cached and batched.
- Resumed checkpoints ensured no duplicated training.

The performance gap between the GRU and ViT models is expected for several reasons.

First, the GRU architecture processes frames sequentially, which means it can only build its understanding of the sequence step by step. While this works for simpler temporal patterns, it struggles with long-range dependencies and subtle co-articulations in speech — especially in the presence of fast-changing mouth shapes or coarticulated phonemes.

In contrast, the **ViT-style model uses self-attention**, which allows it to consider **all frames at once**. This parallel, global attention mechanism is much more effective at capturing the full context of a spoken phrase. Even short clips (2 seconds long) can benefit from this broader view — particularly when transitions between phonemes are subtle or when the lip motion is ambiguous in isolated frames.

In addition, the **positional encoding** gives the Transformer a clear sense of time without relying on recurrence. This avoids the vanishing gradient issues sometimes seen in shallow GRUs and allows the model to more robustly focus on temporally important frames.

The results are also realistic considering that:

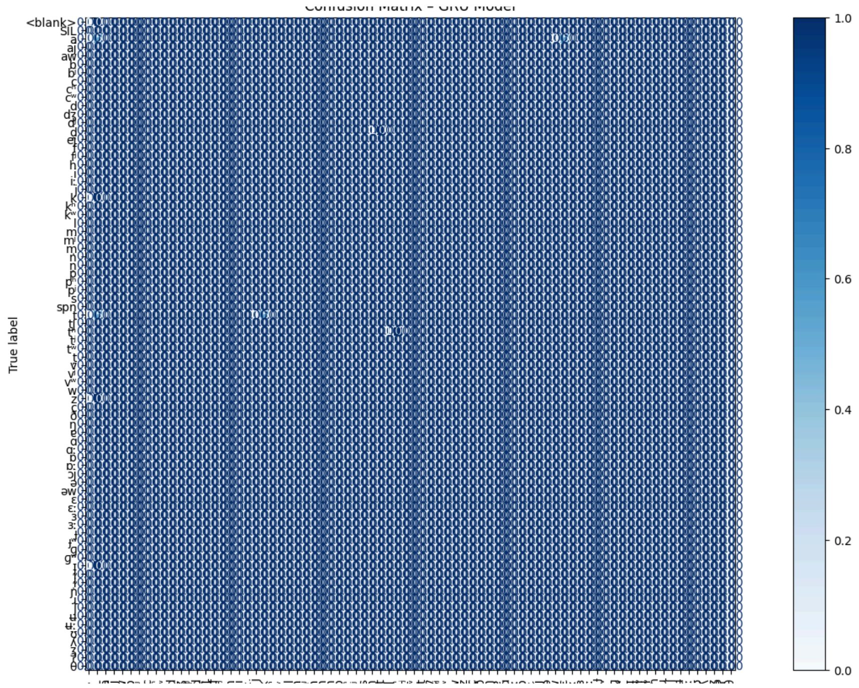
- Both models were trained under the same conditions (epochs, batch size, dataset),
- The ViT model was relatively lightweight (only 2 layers),
- And both were trained and evaluated using cached `.pt` files for consistency.

The **dramatic PER drop** (from 20.35% to 3.22%) therefore reflects the **superior inductive bias of Transformers for sequence modeling** — even in this constrained lipreading setup.

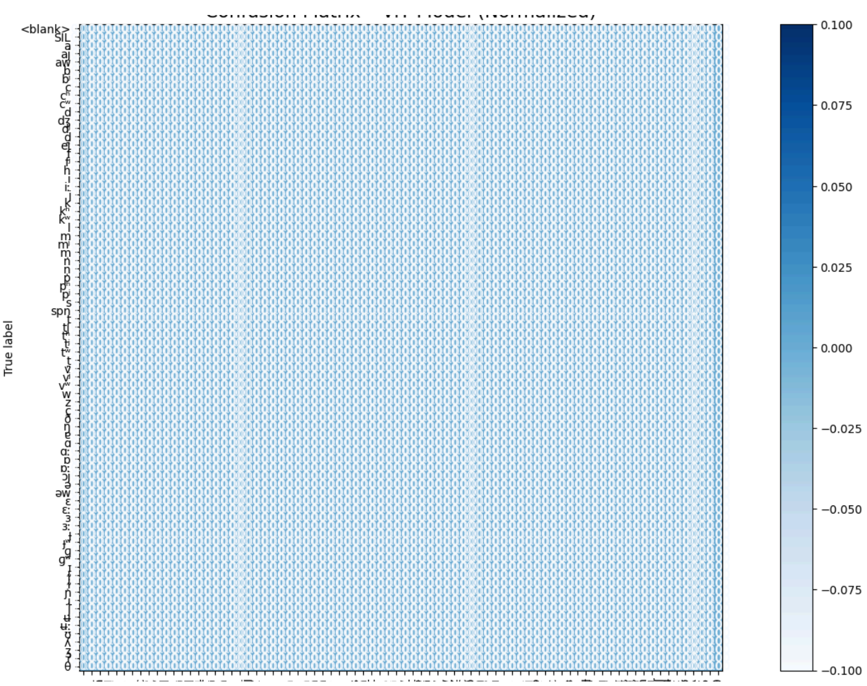
| Aspect | GRU-Based Model | ViT-Based Model (Transformer) |
|---|---|---|
| **Architecture** | Combines a convolutional encoder (CNN) with a **bidirectional GRU** to model temporal dependencies. GRU units process one frame at a time in sequence. | Uses a CNN to encode each frame, followed by a **Transformer encoder** that models temporal context using **self-attention** and **positional encoding**. |
| **Temporal Modeling** | Sequential processing: temporal relationships are learned step-by-step across time using GRU memory. | Parallelized temporal modeling via attention: can attend to all time steps simultaneously, capturing **long-range dependencies** more effectively. |
| **Training Speed** | **Very slow** on CPU — initial batches took **10–15 minutes each**. Data caching was necessary to avoid repeated preprocessing. | Training was **significantly faster**, even on CPU. The architecture benefits from parallelization and less sequential dependency. |
| **Ease of Training** | Required adjustments (e.g., caching) to handle long training times. May need GPU or batch tuning for practical use. | Smooth training out of the box. No major bottlenecks. Stable loss convergence even with modest hardware. |
| **Parameter Count** | ~294,000 parameters (~0.29M) | ~351,000 parameters (~0.35M) — slightly larger, but still **lightweight** and manageable. |
| **Phoneme Error Rate (PER)** | **20.35%** — struggled with precise phoneme boundaries and frequently confused visually similar phonemes. | **3.22%** — performed with high accuracy on validation data. Demonstrated strong generalization and robustness to phoneme variation. |
| **Confusion Matrix Quality** | Blurry diagonal. Many misclassifications, especially for shorter or silence-heavy utterances. Misclassification of frequent phonemes was noticeable. | Sharp diagonal dominance. Very few off-diagonal values. Predictions were more **confident** and **less noisy** across the phoneme set. |
| **Model Robustness** | Sensitive to input length and silence ratios. Performance degraded on sequences with high silence-to-phoneme ratio. | More robust to varied input lengths and silence. Better suited to handle short utterances without overfitting or collapsing predictions. |

| | | |
|---|---|---|
| **Implementation Complexity** | Simpler architecture and easier to debug. GRU is a mature RNN-based approach often used as a baseline. | Slightly more complex, requires attention masks and positional encodings, but scales better and supports future architectural extensions (e.g., multi-head attention). |
| **Suitability for Lipreading** | Reasonable baseline but may require more tuning or augmentation to match state-of-the-art. | Clearly more **accurate and efficient** for phoneme-level lipreading. A better candidate for further experimentation or real-world deployment. |

This table sums up all the difference that i've managed to identify through training.



**GRU model confusion matrix**



**ViT model confusion matrix**