

# Submission Summary

---

## ◆ Task 1 – Long Form ASR Transcription

**1. Overview of the Final Model:** Our final Automatic Speech Recognition (ASR) system is based on Hugging Face Transformers implementation of OpenAI Whisper. We initialized from the pretrained checkpoint `tugstugi/bengaliai-regional-asr_whisper-medium` (769M parameters) and further fine-tuned the model exclusively on the competition dataset to better adapt it to the domain characteristics and long-form speech patterns.

---

## 2. Model Parameter Size

- Base Model: `tugstugi/bengaliai-regional-asr_whisper-medium`
  - Total Parameters: ~769 Million
- 

**3. Inference Time:** Inference was performed on a single T4 GPU (Kaggle environment).

- Total inference time (full test set): ~3 hours 38 minutes
  - Average per audio file (default settings): 8-9 minutes
  - Optimized decoding configuration: 2.5–3 minutes per audio
- 

## 4. Model Artifact

The fine-tuned model checkpoint is publicly available on Hugging Face:

 [https://huggingface.co/bitwisemind/sam\\_15000\\_clean\\_text\\_full\\_model](https://huggingface.co/bitwisemind/sam_15000_clean_text_full_model)

---

**5. Datasets Used:** Only the official competition dataset was used for training and validation.

---

**6. Training Details:** `bengaliAI/tugstugi_bengaliai-regional-asr_whisper-medium` | 90/10 split (1–15229 segments of whole data), batch  $8 \times 2 = 16$ , AdamW 8bit (5e-6), cosine, 6 epochs, fp16 + grad checkpointing, eval/save every 6000 (limit 2). Max 30s audio / 1000 text, augmentation 0.3, gen max 225

---

## 7. Training / Fine-Tuning/ Inference Notebooks

- **GitHub:** [https://github.com/sazzadadib/BitwiseMind\\_DL\\_Sprint\\_4.0/tree/main/Bengali%20Long-form%20Speech%20Recognition](https://github.com/sazzadadib/BitwiseMind_DL_Sprint_4.0/tree/main/Bengali%20Long-form%20Speech%20Recognition)

---

## ◆ Task 2 – Speaker Diarization

**1. Overview of the Final Model:** Our speaker diarization system is built using the pyannote.ai diarization pipeline, powered by the pyannote.audio framework. We used the pretrained segmentation model: **pyannote/segmentation-3.0**. This model detects speaker change boundaries and generates time-stamped speaker segments. The diarization pipeline then clusters segments into distinct speaker identities.

---

## 2. Model Parameter Size

- **Segmentation Model:** pyannote/segmentation-3.0
  - **Total Parameters:** ~1.5 Million
- 

**3. Inference Time:** Inference was performed on a single T4 GPU (Kaggle environment).

- Total inference time (full test set):: ~1 hour 20 minutes
  - Average per audio file: 5–6 minutes
- 

## 4. Model Artifact

The trained diarization checkpoint is available on Hugging Face:

🔗 Finetuned Segmentation Model: [https://huggingface.co/Amanafif554/diarizaaaaaation\\_CKPT\\_segmentation-epoch=39.ckpt](https://huggingface.co/Amanafif554/diarizaaaaaation_CKPT_segmentation-epoch=39.ckpt), was used in inference pipeline.

---

**5. Datasets Used:** Only the official competition dataset was used.

---

**6. Training Details:** pyannote/segmentation-3.0 with pyannote/speaker-diarization-community-1 | 10s chunks, max 3 speakers/chunk (1/frame), 50 epochs on 1 GPU (Lightning). Monitor loss/train (min), save best+last, metric DER, protocol CustomData.SpeakerDiarization.train with train/dev/test splits.

---

## 7. Training / Fine-Tuning/ Inference Notebooks

- **GitHub:** [https://github.com/sazzadadib/BitwiseMind\\_DL\\_Sprint\\_4.0/tree/main/Bengali%20Speaker%20Diarization](https://github.com/sazzadadib/BitwiseMind_DL_Sprint_4.0/tree/main/Bengali%20Speaker%20Diarization)