

```
import logging
```

```
from faster_whisper import WhisperModel
```

```
class FasterWhisperTranscriber:
```

```
    def __init__(
```

```
        self,
```

```
        model_size="large-v3",
```

```
        device="cuda",
```

```
        compute_type="float16",
```

```
        model_type="faster-whisper",
```

```
        **kwargs,
```

```
    ):
```

```
        """
```

```
        Initialize the WhisperModel with specified configuration.
```

```
        :param model_size: Size of the Whisper model (e.g., 'large-v3', 'distil-large-v2')
```

```
        :param device: Computation device ('cuda' or 'cpu')
```

```
        :param compute_type: Type of computation ('float16', 'int8_float16', 'int8')
```

```
        :param model_type: Type of model ('faster-whisper' or 'faster-distil-whisper')
```

```
        :param kwargs: Additional arguments for WhisperModel transcribe method
```

```
        """
```

```
        self.model_size = model_size
```

```
        self.device = device
```

```
        self.compute_type = compute_type
```

```
        self.model_type = model_type
```

```

self.transcribe_options = kwargs

self.model = WhisperModel(
    self.model_size, device=self.device, compute_type=self.compute_type
)

def run(self, task: str, *args, **kwargs):
    """
    Transcribes the given audio file using the Whisper model.

    :param audio_file_path: Path to the audio file to be transcribed
    :return: Transcription results
    """
    segments, info = self.model.transcribe(task, **self.transcribe_options)

    # Printing language detection information
    print(
        f"Detected language '{info.language}' with probability {info.language_probability:.2f}"
    )

    # Handling transcription based on the model type
    if self.model_type == "faster-whisper":
        for segment in segments:
            print(f"[{segment.start:.2fs} -> {segment.end:.2fs}] {segment.text}")
    elif (
        self.model_type == "faster-distil-whisper"
        and "word_timestamps" in self.transcribe_options
    )

```

```

        and self.transcribe_options["word_timestamps"]

    ):

        for segment in segments:

            for word in segment.words:

                print(f"[{word.start:.2fs} -> {word.end:.2fs}] {word.word}")

            else:

                for segment in segments:

                    print(f"[{segment.start:.2fs} -> {segment.end:.2fs}] {segment.text}")

```

Example usage

```

if __name__ == "__main__":

    logging.basicConfig()

    logging.getLogger("faster_whisper").setLevel(logging.DEBUG)

```

Example for faster-whisper with GPU and FP16

```

transcriber = FasterWhisperTranscriber(

    model_size="large-v3", device="cuda", compute_type="float16", beam_size=5

)

transcriber.run("song.mp3")

```