Отчет по Заданию №7

Юй Чанбай

14 май 2025г.

1 Цель работы

Изучайте когнитивные характеристики мозга на основе данных ЭЭГ.В частности, необходимо обработать набор данных ЭЭГ (электроэнцефалограммы), чтобы решить проблему двух классификаций (например, сжатие левого кулака или правого кулака).При анализе набора данных непрерывное вейвлет-преобразование (CWT) и кратковременное преобразование Фурье (STFT) используются для преобразования временного сигнала ЭЭГ, а затем гибридная модель CNN-2D + LSTM и модель CNN-2D используются для обучения классификации.

2 Экспериментальный процесс

2.1 Обработка данных

Набор данных

Тренировочные данные: 400 образцов; тестовые данные: 320 образцов; целевая маркировка: 2 категории (1-левая рука, 2-правая рука).

```
# download dataset
x_train = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/MI-EEG-S9T.csv?raw=true",header=None)
x_test = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/MI-EEG-S9E.csv?raw=true",header=None)
y_train = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/2class_MI_EEG_train_9.csv?raw=true",header=None)
y_test = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/2class_MI_EEG_test_9.csv?raw=true",header=None)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(400, 3000)
(320, 3000)
(400, 1)
(320, 1)
```

1 目标

基于脑电图数据研究大脑认知特征。具体而言，需要处理 EEG（脑电图）数据集，以解决二分类问题（例如握紧左拳或右拳）。分析数据集时需使用连续小波变换(CWT)与短时傅里叶变换(STFT)对时间脑电信号进行变换，之后分别使用 CNN-2D + LSTM 混合模型和 CNN-2D 模型进行分类的训练。

2 实验流程

2.1 数据处理

数据集

训练数据：400 个样本；测试数据：320 个样本；目标标签:2 类(1-左手,2-右手).

```
# download dataset
x_train = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/MI-EEG-S9T.csv?raw=true",header=None)
x_test = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/MI-EEG-S9E.csv?raw=true",header=None)
y_train = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/2class_MI_EEG_train_9.csv?raw=true",header=None)
y_test = pd.read_csv("https://github.com/TAUforPython/BioMedAI/blob/main/test_datasets/2class_MI_EEG_test_9.csv?raw=true",header=None)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(400, 3000)
(320, 3000)
(400, 1)
(320, 1)
```

Предварительная обработка сигналов

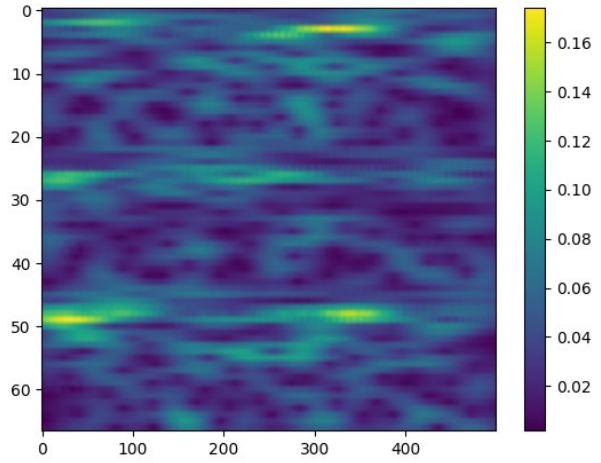

Рис1.Непрерывное вейвлет-преобразование (CWT)

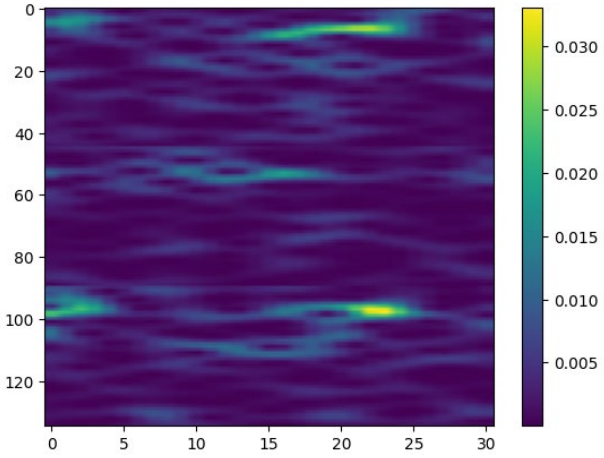

Рис2.Преобразование Фурье (STFT)

信号预处理

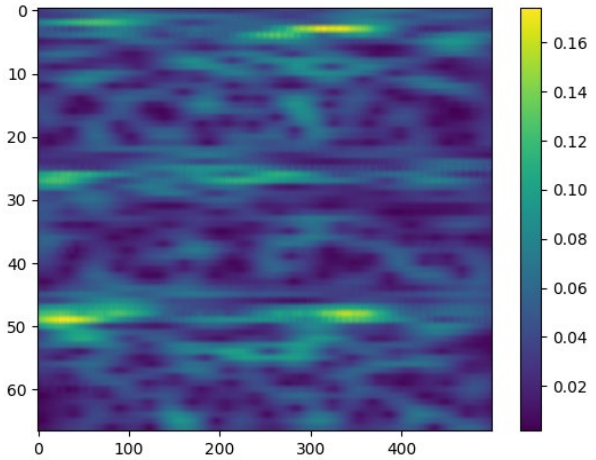

图 1. 连续小波变换(CWT)

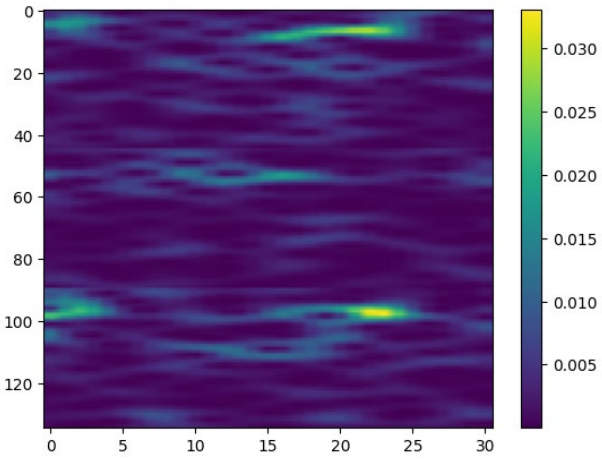

图 2. 短时傅里叶变换(STFT)

2.2 Построение нейронной сети

Вейвлет-преобразование с использованием сети CNN-2D. Конкретная сеть состоит из 2 сверточных слоев (4 фильтра, 3× 3 ядра), самого большого пула (2x2) и полностью подключенного слоя (32 нейрона). отсев=0,5.Архитектура модели показана на рисунке 3:

2.2 构建神经网络

连续小波变换使用 CNN-2D 网络，具体网络为 2 卷积层（4 个滤波器，3×3 核），最大池（2x2），全连接层（32 个神经元），dropout=0.5。模型架构如图 3 所示：

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 67, 500, 4)	40
max_pooling2d_8 (MaxPooling2D)	(None, 33, 250, 4)	0
conv2d_9 (Conv2D)	(None, 33, 250, 4)	148
max_pooling2d_9 (MaxPooling2D)	(None, 16, 125, 4)	0
flatten_4 (Flatten)	(None, 8000)	0
dense_8 (Dense)	(None, 32)	256,032
dropout_4 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 2)	66

Total params: 768,860 (2.93 MB)
Trainable params: 256,286 (1001.12 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 512,574 (1.96 MB)

Рисунок 3. Сетевая архитектураCNN-2D

Model: "sequential_4"

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 67, 500, 4)	40
max_pooling2d_8 (MaxPooling2D)	(None, 33, 250, 4)	0
conv2d_9 (Conv2D)	(None, 33, 250, 4)	148
max_pooling2d_9 (MaxPooling2D)	(None, 16, 125, 4)	0
flatten_4 (Flatten)	(None, 8000)	0
dense_8 (Dense)	(None, 32)	256,032
dropout_4 (Dropout)	(None, 32)	0
dense_9 (Dense)	(None, 2)	66

Total params: 768,860 (2.93 MB)
Trainable params: 256,286 (1001.12 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 512,574 (1.96 MB)

图 3. CNN-2D 网络

Для кратковременного преобразования Фурье используется сеть CNN-2D+LSTM. Конкретными сетями являются: TimeDistributed (слой, используемый для обработки синхронизации), LSTM (4 нейрона) и полностью подключенный слой (32 нейрона).

Model: "sequential_4"

Layer (type)	Output Shape	Param #
time_distributed_20 (TimeDistributed)	(None, 1, 135, 31, 4)	40
time_distributed_21 (TimeDistributed)	(None, 1, 67, 15, 4)	0
time_distributed_22 (TimeDistributed)	(None, 1, 67, 15, 4)	148
time_distributed_23 (TimeDistributed)	(None, 1, 33, 7, 4)	0
time_distributed_24 (TimeDistributed)	(None, 1, 924)	0
lstm_4 (LSTM)	(None, 4)	14,864
dense_8 (Dense)	(None, 32)	160
dense_9 (Dense)	(None, 2)	66

Total params: 45,836 (179.05 KB)
Trainable params: 15,278 (59.68 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 30,558 (119.37 KB)

Рисунок 4. Сетевая архитектураCNN-2D+LSTM

短时傅里叶变换使用 CNN-2D+LSTM 网络具体网络为：TimeDistributed（用于处理时序的图层）， LSTM（4 个神经元），全连接层（32 个神经元）。

Model: "sequential_4"

Layer (type)	Output Shape	Param #
time_distributed_20 (TimeDistributed)	(None, 1, 135, 31, 4)	40
time_distributed_21 (TimeDistributed)	(None, 1, 67, 15, 4)	0
time_distributed_22 (TimeDistributed)	(None, 1, 67, 15, 4)	148
time_distributed_23 (TimeDistributed)	(None, 1, 33, 7, 4)	0
time_distributed_24 (TimeDistributed)	(None, 1, 924)	0
lstm_4 (LSTM)	(None, 4)	14,864
dense_8 (Dense)	(None, 32)	160
dense_9 (Dense)	(None, 2)	66

Total params: 45,836 (179.05 KB)
Trainable params: 15,278 (59.68 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 30,558 (119.37 KB)

图 4. CNN-2D+LSTM 网络

2.3 Обучение и оценка модели (5 итераций)

2. 3 训练并评估模型（5 次迭代）

Результат CWT+CNN-2d выглядит следующим образом

```
Resultados:
loss: [0.4424828886985779, 0.4719349443912506, 0.5195335149765015, 0.49684199690818787, 0.622015118598938]
accuracy: [0.800000011920929, 0.765625, 0.765625, 0.778124988079071, 0.71875]
kappa: [np.float64(0.4375)]
time_elapsed: 84.51700043678284
```

Рисунок 5.Общие результаты (CWT+CNN-2d)

CWT+CNN-2d 结果如下所示

Resultados:
loss: [0.4424828886985779, 0.4719349443912506, 0.5195335149765015, 0.49684199690818787, 0.622015118598938]
accuracy: [0.800000011920929, 0.765625, 0.765625, 0.778124988079071, 0.71875]
kappa: [np.float64(0.4375)]
time_elapsed: 84.51700043678284

图 5. 总结果(CWT+CNN-2d)

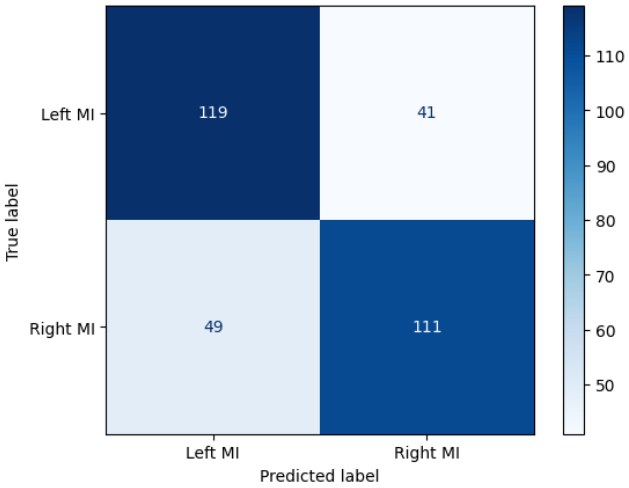

Рисунок 6.Матрица путаницы (CWT+CNN-2d)

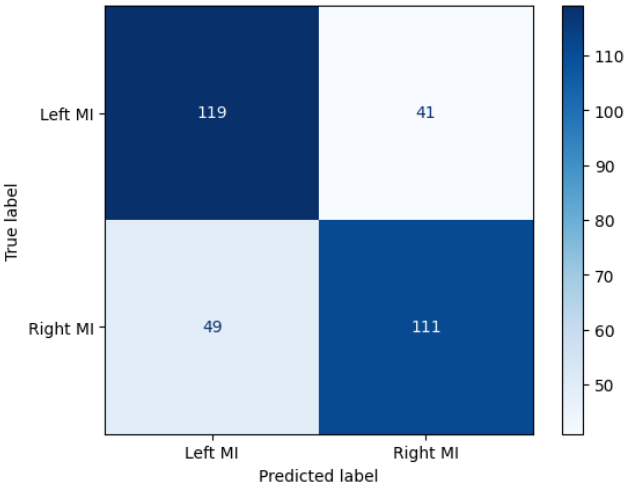

图 6. 混淆矩阵(CWT+CNN-2d)

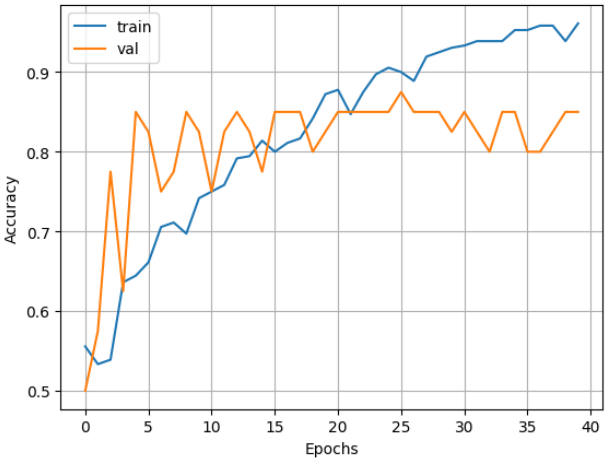

Рисунок 7.Точность (CWT+CNN-2d)

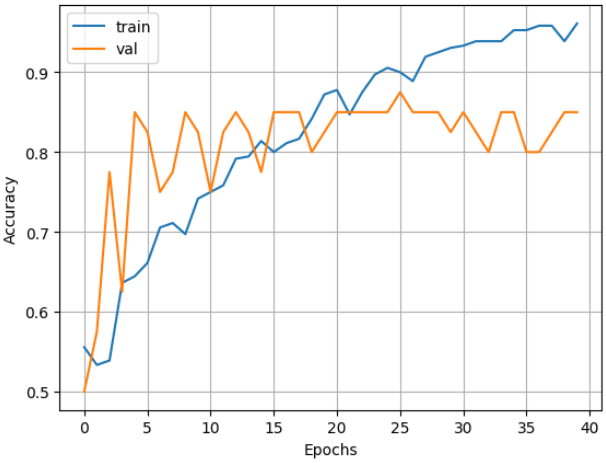

图 7. 准确性(CWT+CNN-2d)

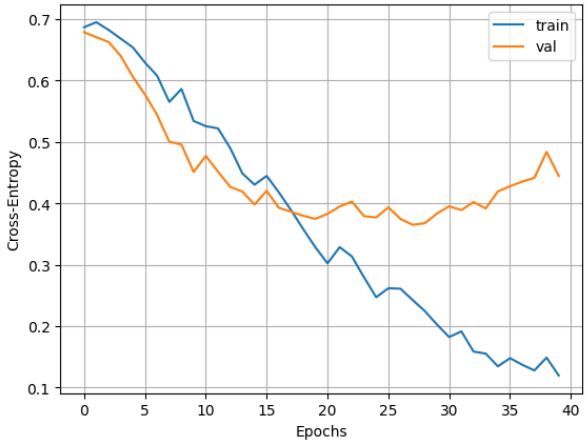

Рисунок 8.Перекрестная энтропия (CWT+CNN-2d)

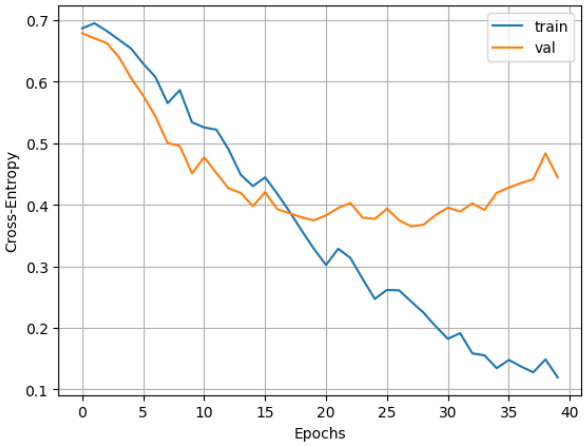

图 8. 交叉熵(CWT+CNN-2d)

Результат STFT+CNN-2D + LSTM выглядит следующим образом

STFT+CNN-2D + LSTM 结果如下所示

```
Resultados:
loss: [0.47591632604599, 0.5104565024375916, 0.4850081503391266, 0.473102867603302, 0.4563733637332916]
accuracy: [0.778124988079071, 0.746874988079071, 0.768750011920929, 0.784375011920929, 0.784375011920929]
kappa: [np.float64(0.525)]
time_elapsed: 91.38264560699463
```

Рисунок 9.Общие результаты (STFT+ CNN-2D + LSTM)


```
Resultados:
loss: [0.47591632604599, 0.5104565024375916, 0.4850081503391266, 0.473102867603302, 0.4563733637332916]
accuracy: [0.778124988079071, 0.746874988079071, 0.768750011920929, 0.784375011920929, 0.784375011920929]
kappa: [np.float64(0.525)]
time_elapsed: 91.38264560699463
```


图 9. 总结果 (STFT+CNN-2D + LSTM)

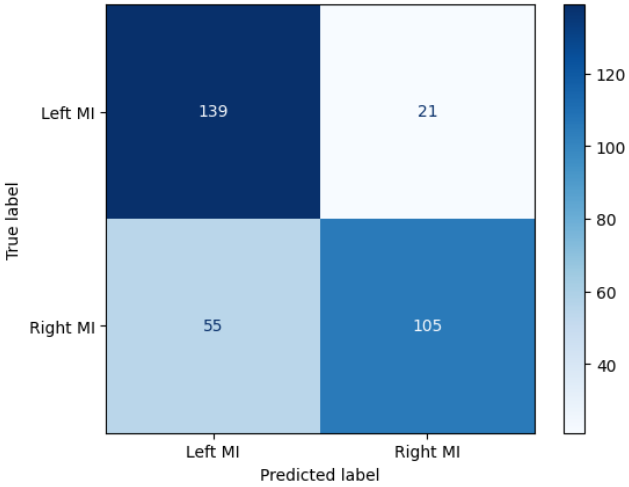

Рисунок 10.Матрица путаницы (STFT+ CNN-2D + LSTM)

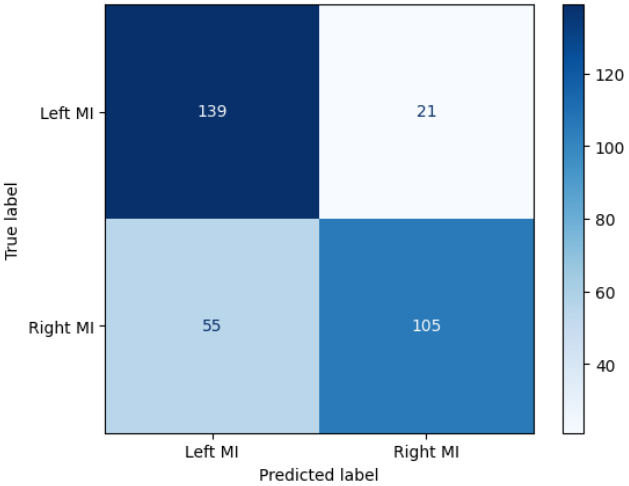

图 10. 混淆矩阵 (STFT+CNN-2D + LSTM)

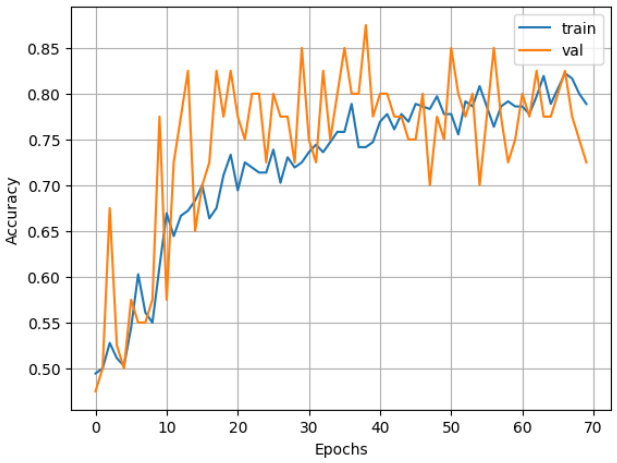

Рисунок 11.Точность (STFT+ CNN-2D + LSTM)

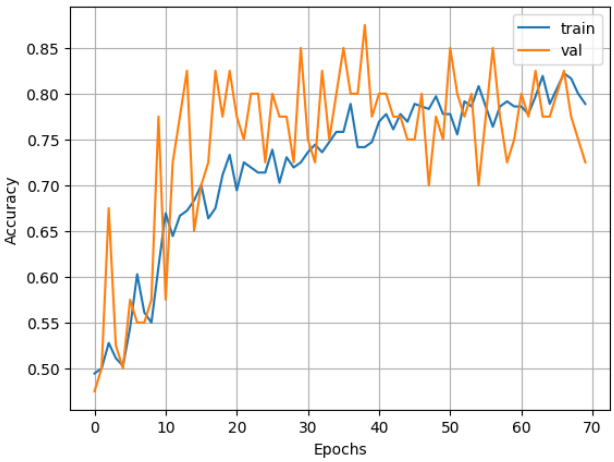

图 11. 准确性 (STFT+CNN-2D + LSTM)

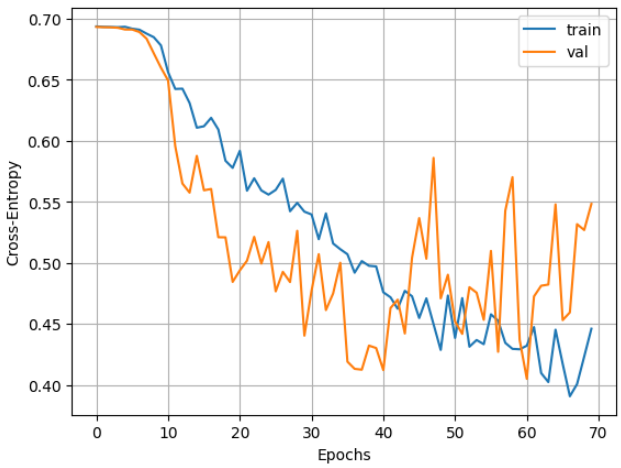

Рисунок 12.Перекрестная энтропия

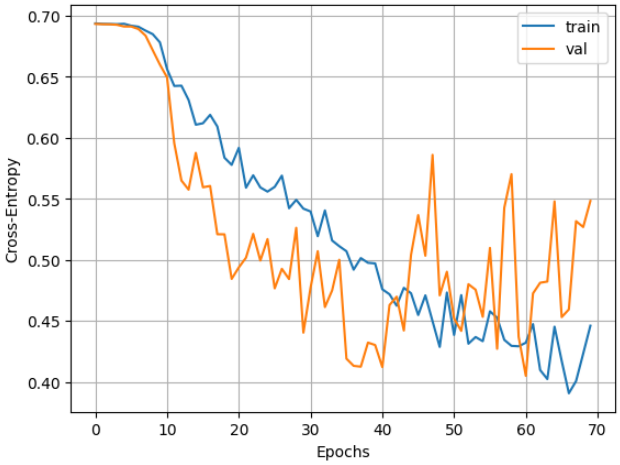

图 12. 交叉熵 (STFT+CNN-2D + LSTM)