Report with potential improvements

Task 1

The first task is to train a NER model to identify mountain names inside the texts.

Methodology

Dataset creation

To create a dataset for model training, I asked ChatGPT to create two lists:

- 100 mountain names
- 192 sentences of different lengths with context about mountains

After that, I formed a dataset similar to the WNUT 17. In my case, it is JSON with such fields as “sentence”, “tokens”, and “ner_tags”. NER_tags are in BIO format.

- B-MOUNTAIN - beginning of mountain name.
- I-MOUNTAIN - inside of the entity.
- O - outside of the entity.

Task solving

I used an article from HuggingFace to solve this task, as suggested by my lecturer at a recent lecture in my course “Natural Language Processing” at my university.

I used the DistilBERT base model (uncased), the solution is more explained in README.md in folder Task_1. I divided data into train and validation datasets and trained this model for 5 epochs.

Results

Inference for text = "Next on our list is Denali Peak, also known as Mount McKinley, in Alaska."


```
[{'entity': 'B-MOUNTAIN', 'score': 0.49926996, 'index': 6, 'word': 'den', 'start': 20, 'end': 23}, {'entity': 'I-MOUNTAIN', 'score': 0.5477206, 'index':
7, 'word': '##ali', 'start': 23, 'end': 26}, {'entity': 'B-MOUNTAIN', 'score': 0.84802276, 'index': 8, 'word': 'peak', 'start': 27, 'end': 31}, {'entit
y': 'B-MOUNTAIN', 'score': 0.99523646, 'index': 13, 'word': 'mount', 'start': 47, 'end': 52}, {'entity': 'I-MOUNTAIN', 'score': 0.94849247, 'index': 14,
'word': 'mckinley', 'start': 53, 'end': 61}]
```


Den	B-MOUNTAIN
ali	I-MOUNTAIN
Peak	B-MOUNTAIN
Mount	B-MOUNTAIN
McKinley	I-MOUNTAIN

This sentence wasn't in the dataset, but these mountains were, so in this case, the model showed great results. "Denali" was divided into tokens "Den" and "ali", but they are still considered the same mountain. The only problem is that "Peak" is labelled as a separate mountain.

These mountains and sentences are from Wikipedia, not in the dataset, the model still performs well, but it doesn't recognise the whole name and divides mountain names into pieces.
text = "The highest peak is Thabana Ntlenyana, at 3,482 m (11,424 ft)."

```
Tha             B-MOUNTAIN
bana            B-MOUNTAIN
Nt              I-MOUNTAIN
len             I-MOUNTAIN
yana            I-MOUNTAIN
```

text = "Therefore, this portion of escarpment is not so impressive as the Mpumalanga and Lesotho stretches of the Drakensberg."

```
Drake           B-MOUNTAIN
```

Discussion

Ways to improve the model:

- **Find approaches** so that **mountains not in the dataset will be identified correctly** without subword division, as it is the main problem.
- **Enlarge dataset with sentences from real texts.** It will allow the model to identify mountain names in more different contexts, and it will perform better on real data. It is a lot of work, as sentences should be annotated, but it will boost model performance.
- **Try different models for this task.** The model trained on conll2003 data may perform better, but it already identifies mountains as locations. Still, experiments will show if the performance becomes better.
- **Experiment with parameters.** Bigger datasets will probably require longer training. There can be experiments with epochs, batch_size, learning_rate, etc. Also, I didn't use any optimizers and schedulers, but they can boost model performance.
- **Try different approaches.** Spacy library and LLM zero-shot can also be used for this task. I didn't research these approaches, but their performances can also be compared.

Task 2

The second task is about matching images from different seasons.

Methodology

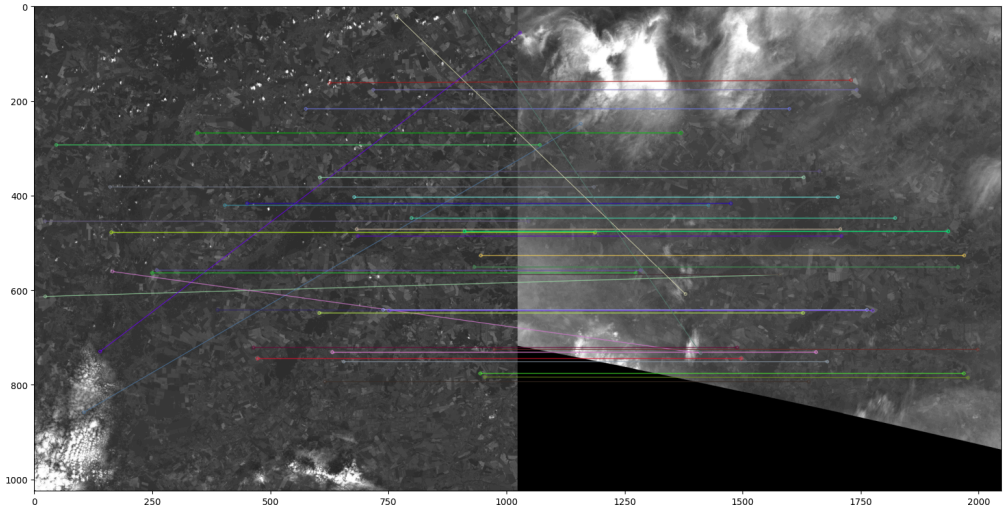
Dataset creation

For this task, I downloaded dataset from Kaggle and parsed it to get photos *_TCI.jpg with images to create a dataset with just photos.

Task solving

I used classical CV approach for this task: SIFT for feature extraction and FLANN for feature matching. Solving explanation is in README.md in folder Task_2.

Results



These photos are pretty similar, and they are from the same season, but still, there is a difference. The classical approach worked well for this pair, but still could be better)

Discussion

- **Deep learning models.** As I used the classical approach, it does not perform well when matching images from different seasons. It would be great to try SuperPoint and SuperGlue, deep learning models.
- **Experiment with parameters.** The parameters trees and checks in FLANN can be increased. Also, I could also experiment with a matching ratio, as I used 0.7, which is common.
- **Try some different image scaling approaches.** I just scaled the image with keeping, but it would be good to research more techniques which will allow to keep more information in the photo.