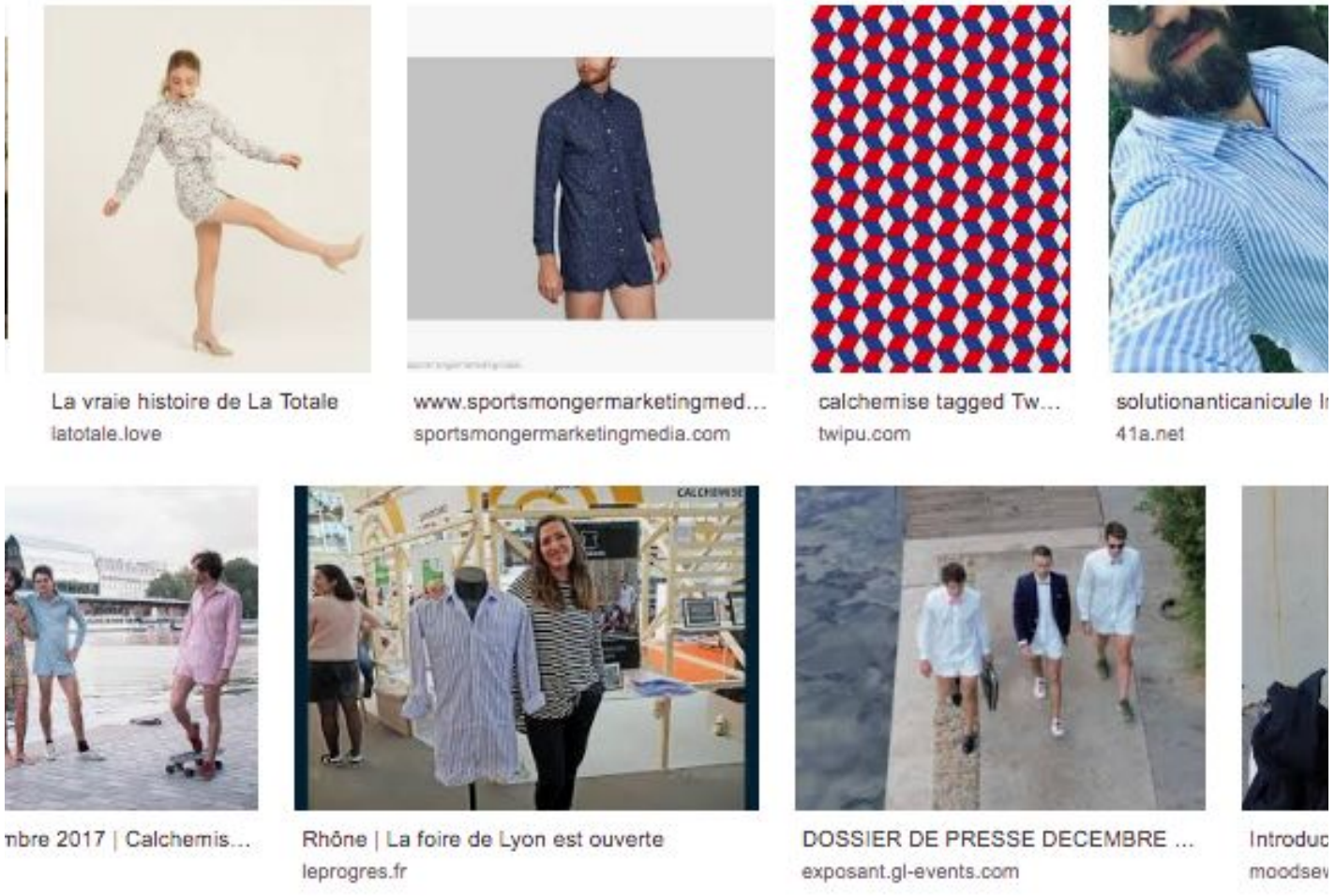
Few shot learning in Production

Few shot learning en production
QUESAKO

k (1~10) reference images per label



n real life images to classify

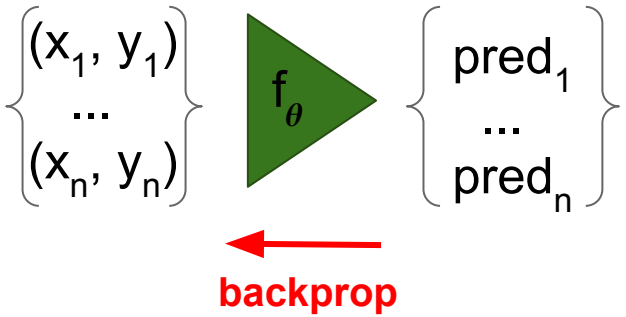
SOMMAIRE

SOMMAIRE

Concept
LEARNING / META LEARNING

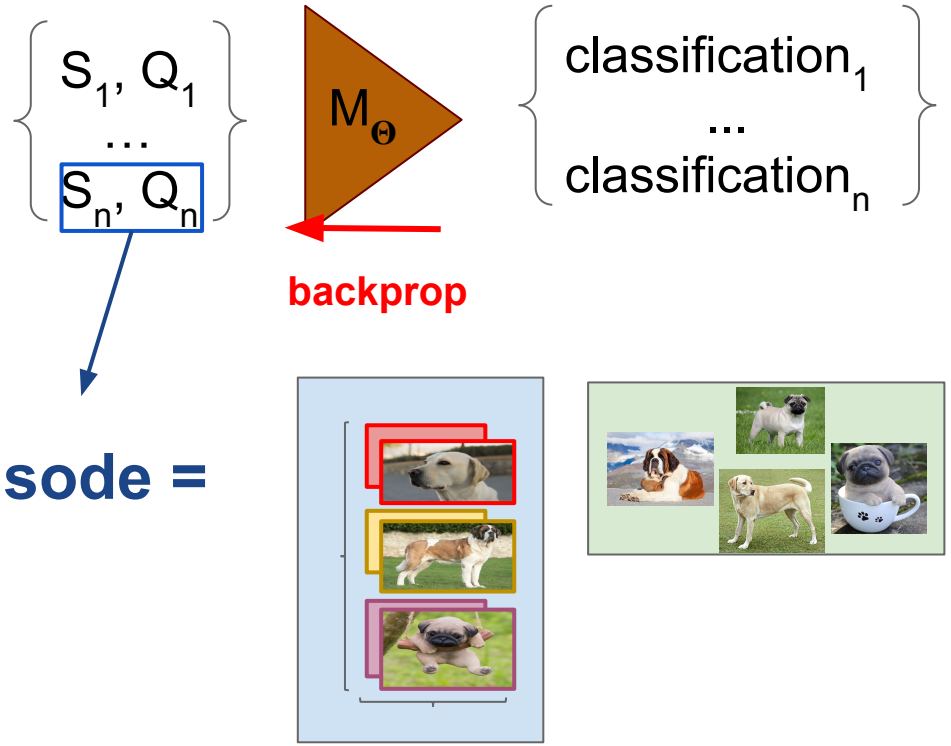
Learning : learn *image* ⟼ *label*

Training epoch



Meta-learning : learn *support-set* ⟼ *(image* ⟼ *label)*

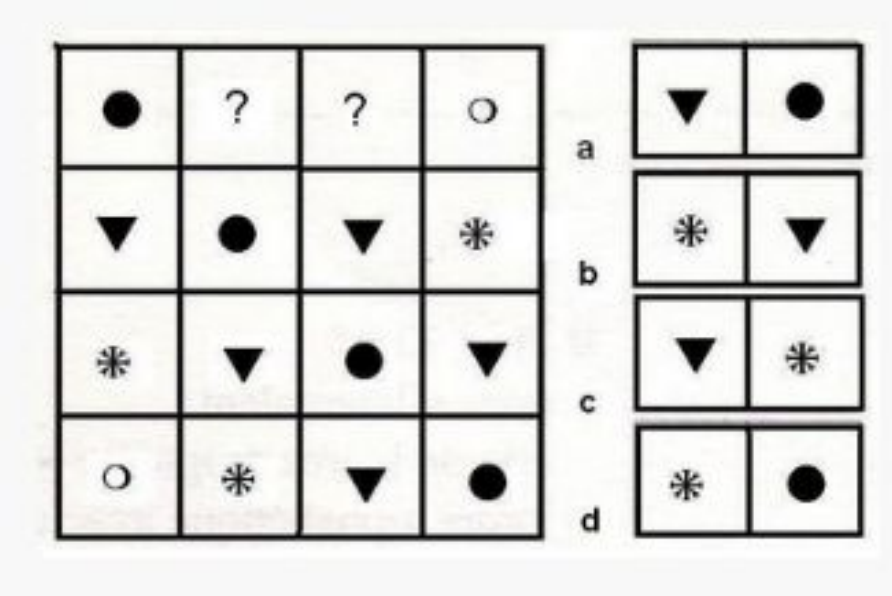
Training epoch



Concept

META-LEARNING : TWO APPROACHES

Without retraining

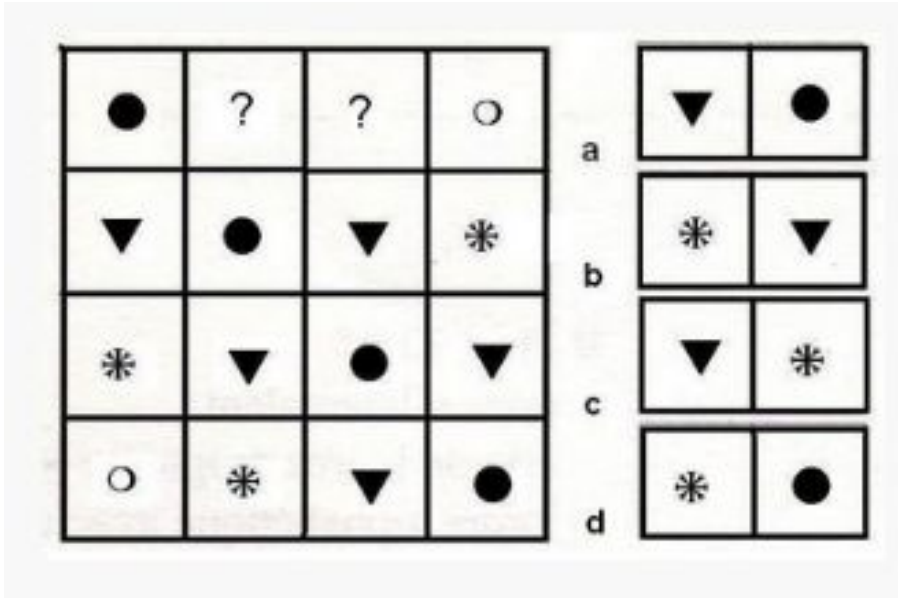


With retraining

Concept

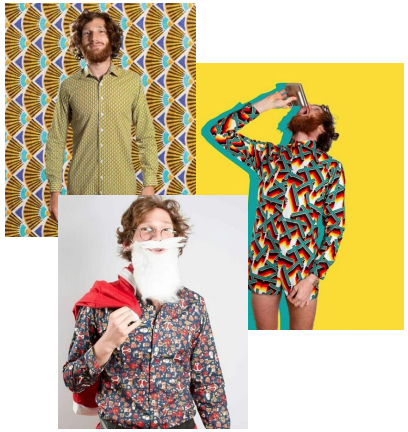
META-LEARNING : DEUX POSSIBILITÉS

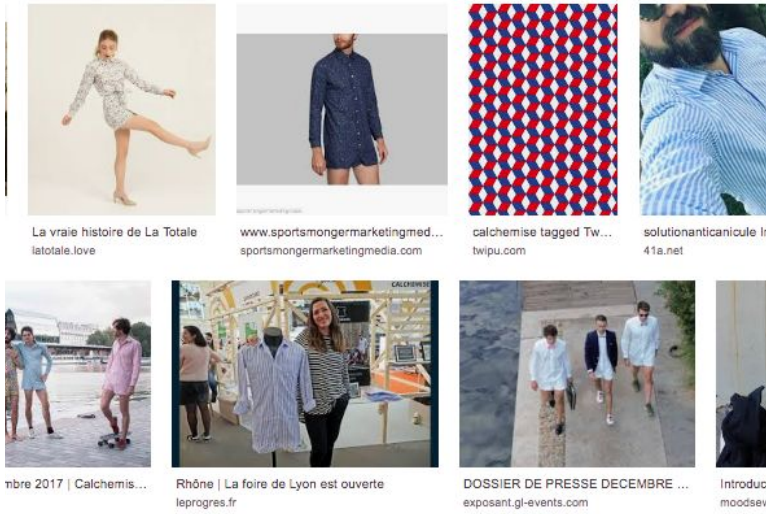
Without retraining

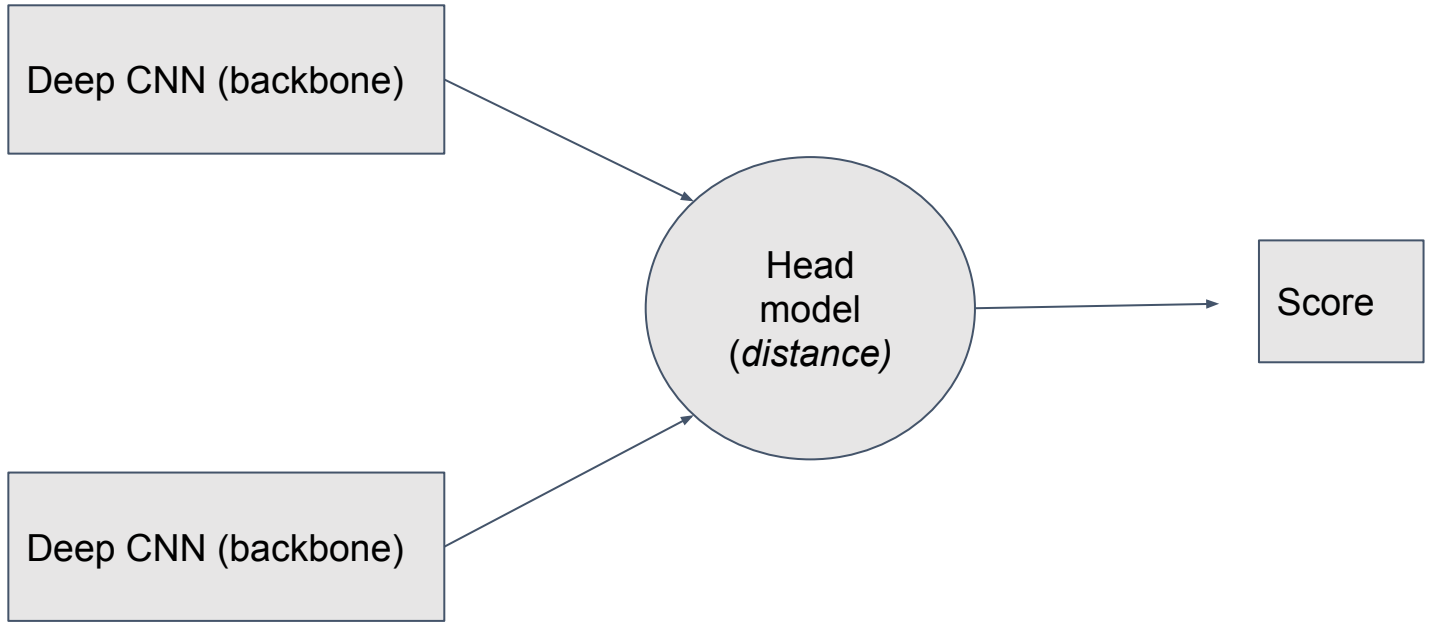


With retraining

Concept
METRIC LEARNING





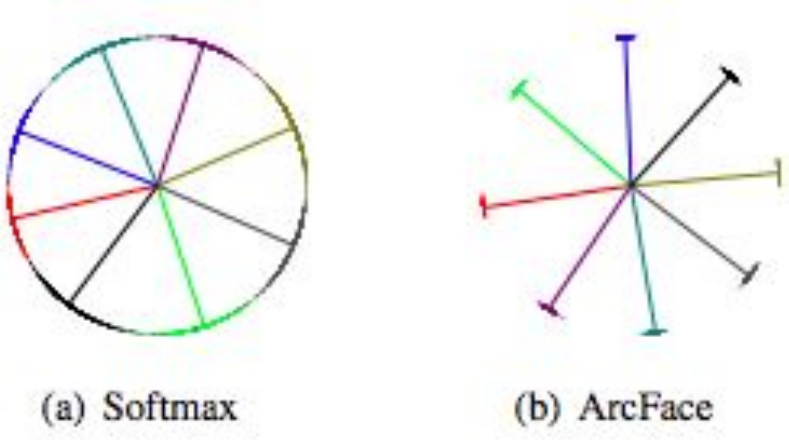


Introduction
LITERATURE

- [A closer look at few shot learning, *Chen et al.*] **(ICLR 19')**
  - *"deeper backbones significantly reduce the performance differences among methods"*

- [Few-Shot Learning with Localization in Realistic Settings, *Wertheimer & Hariharan*, 19']
  - *"We show that prior methods designed for few-shot learning do not work out of the box in real world recognition problems"*

- **Face recognition problem**
  - Similar but with a lot of training data
  - SotA : distance learning.
    - *ArcFace: Additive Angular Margin Loss for Deep Face Recognition, Deng et al. 19'*



(a) Softmax (b) ArcFace

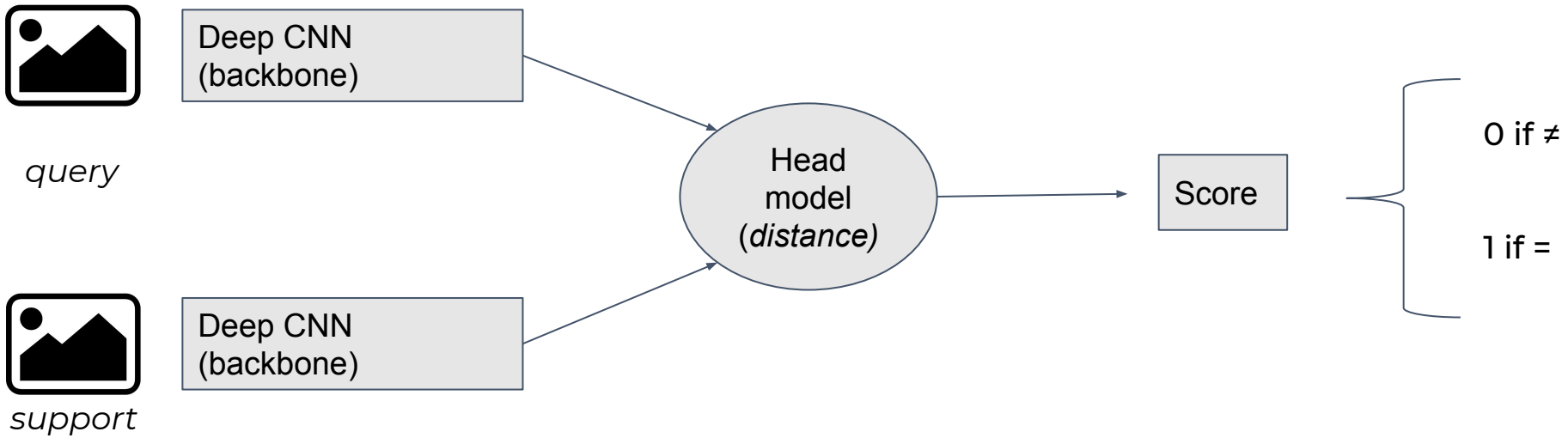
Introduction
PRACTICAL CONSIDERATIONS

	Academic research	**Production**
Dataset	Balanced	Random
Initial weights	Random	Pre-trained (ex. ImageNet)
Dataset size	Constant	Increasing
Support set	Constant	Dynamic
Add of a new class	Theoretical	Instantaneous

SOMMAIRE

Siamese Nets
TRAINING

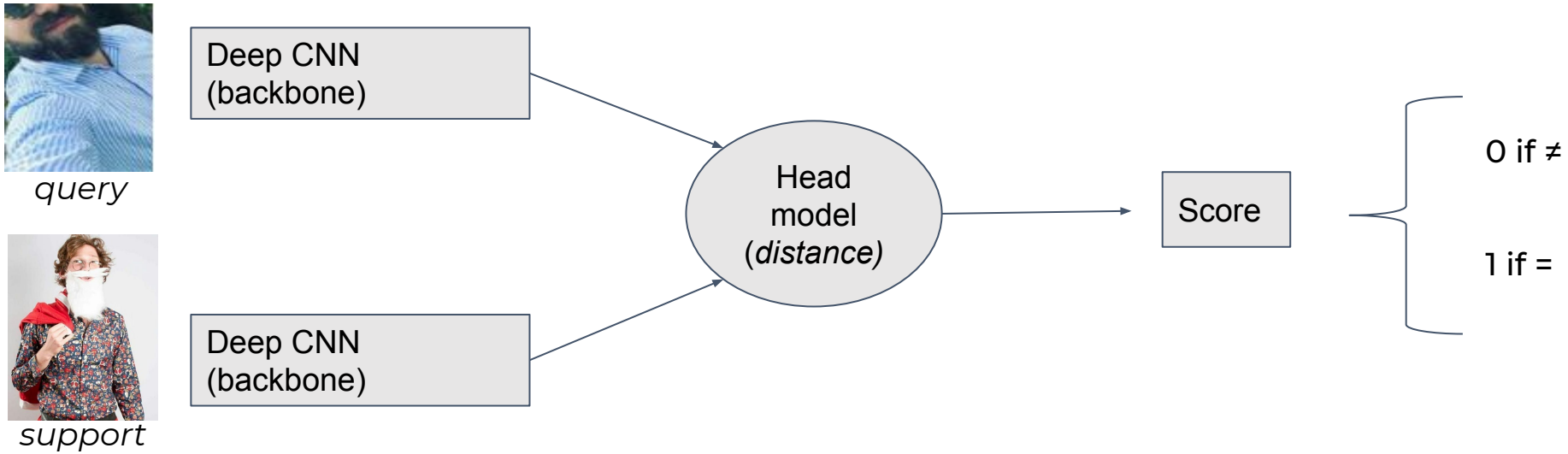
- *Siamese Neural Networks for One-shot Image Recognition, Koch et al. 15'*
- **Architecture**



- **Training**
  - generation of pairs of same and different classes
- **Prediction**
  - use the learnt *metric* to in any usual classifier in the feature space
    - e.g: nearest neighbor (usual Siamese nets)
    - nearest prototypes (ProtoNets)

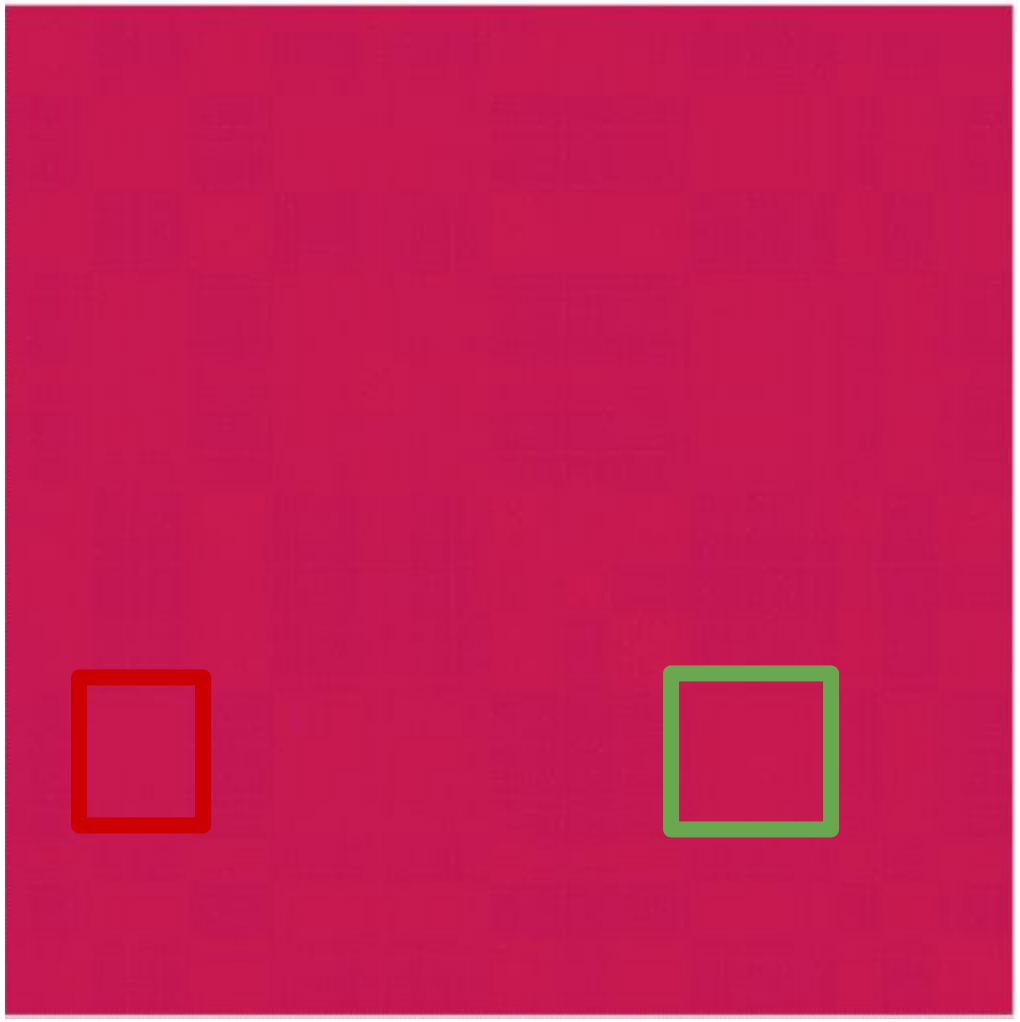
Siamese Nets
TRAINING

- *Siamese Neural Networks for One-shot Image Recognition, Koch et al. 15'*
- **Architecture**



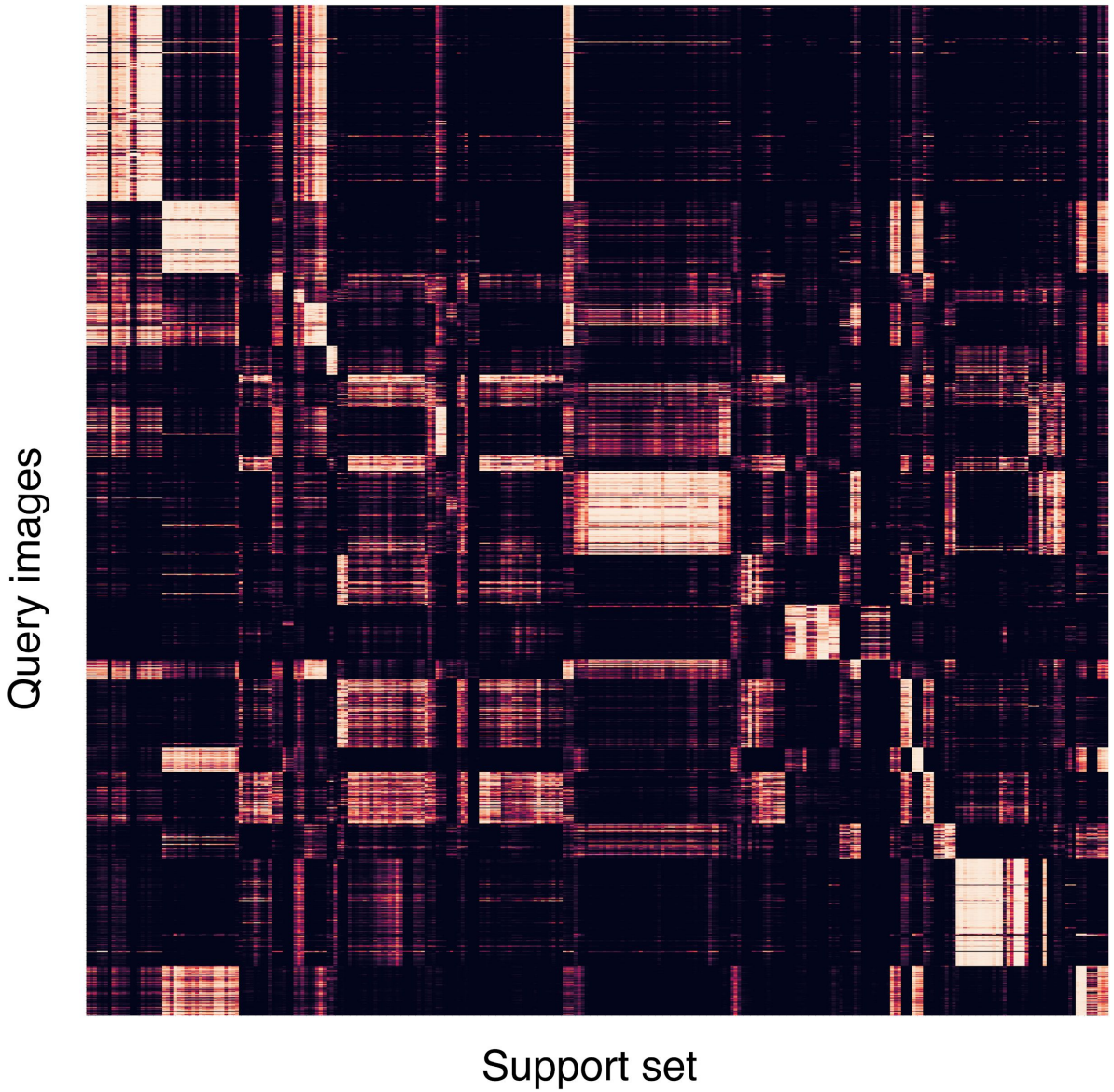
- **Training**
  - generation of pairs of same and different classes
- **Prediction**
  - use the learnt *metric* to in any usual classifier in the feature space
    - e.g: nearest neighbor (usual Siamese nets)
    - nearest prototypes (ProtoNets)

Siamese Nets
VISUALISATION



pair-wise distance matrix over the train set during training; classes are visibles as blocks

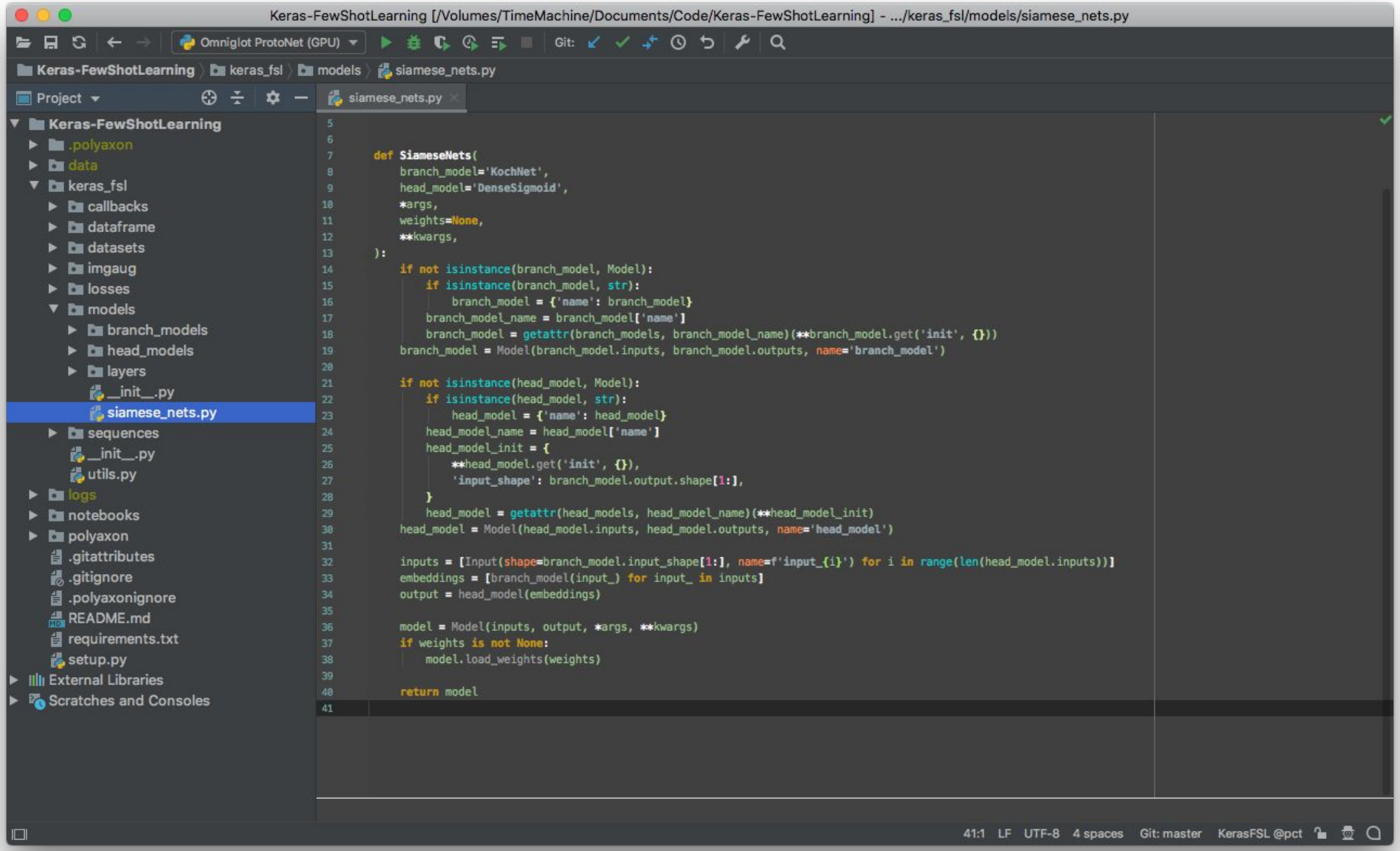
Siamese Nets
VISUALISATION



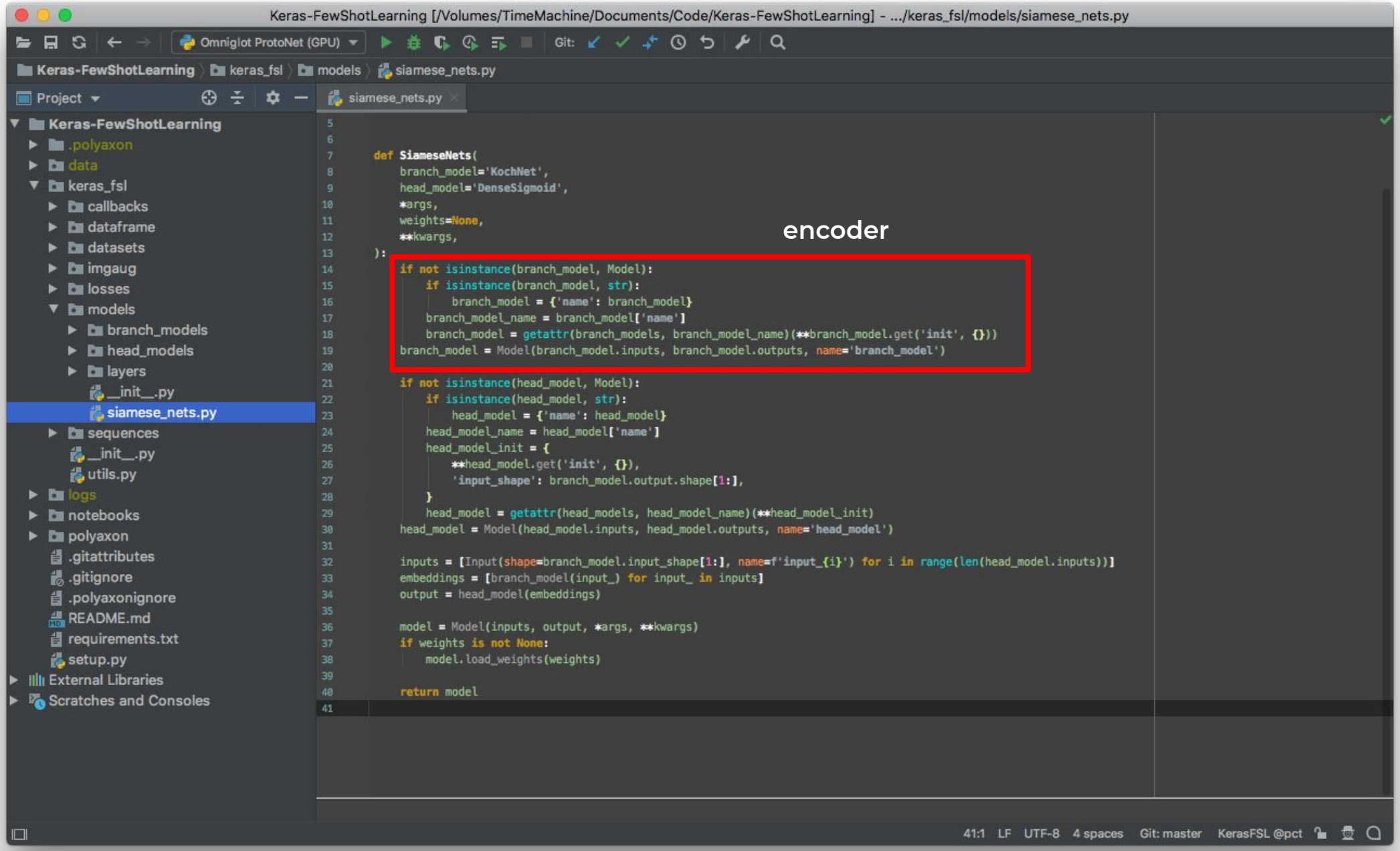
pair-wise distance matrix test set against support set; unsorted query images

SOMMAIRE

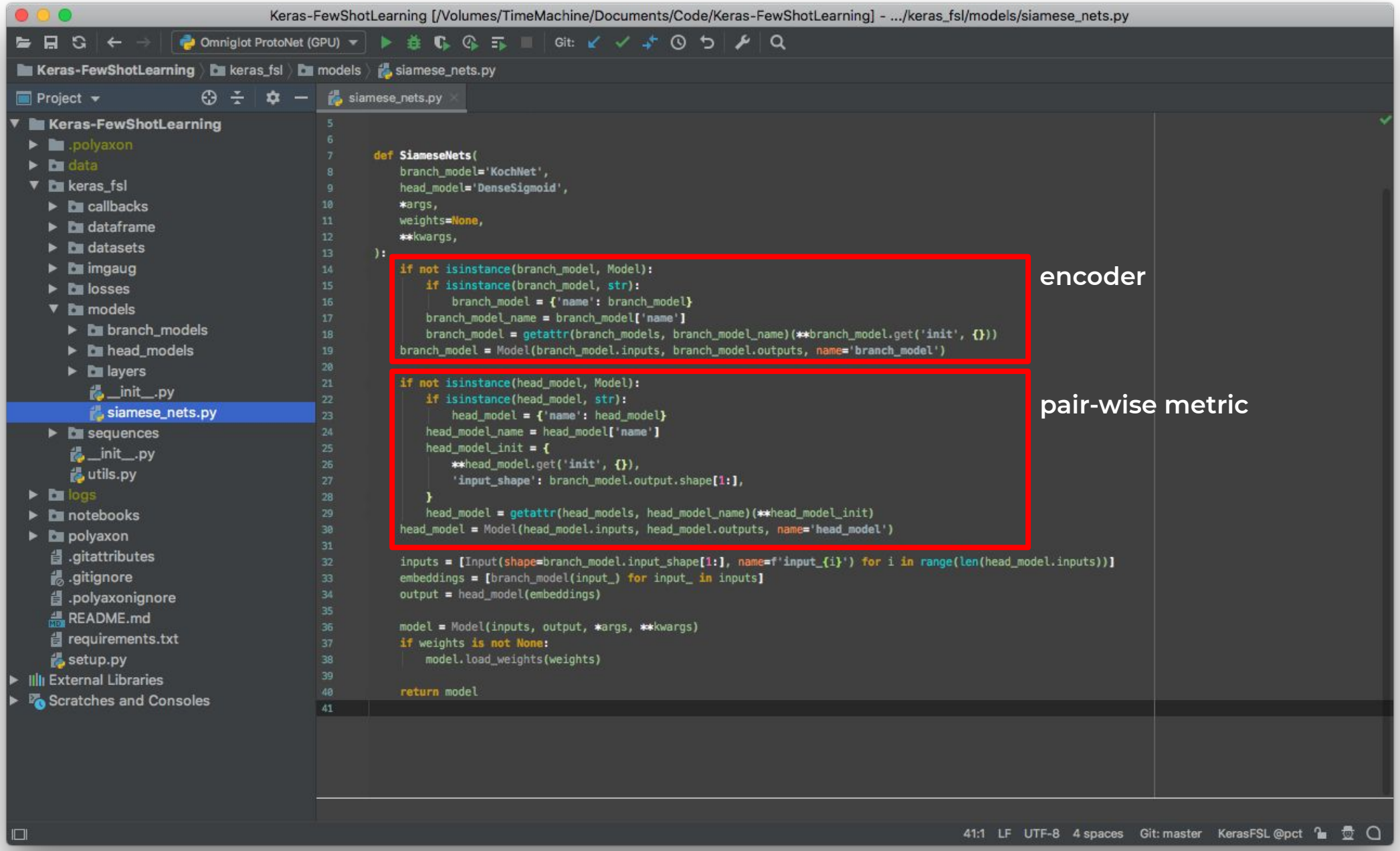
Code
keras_fsl



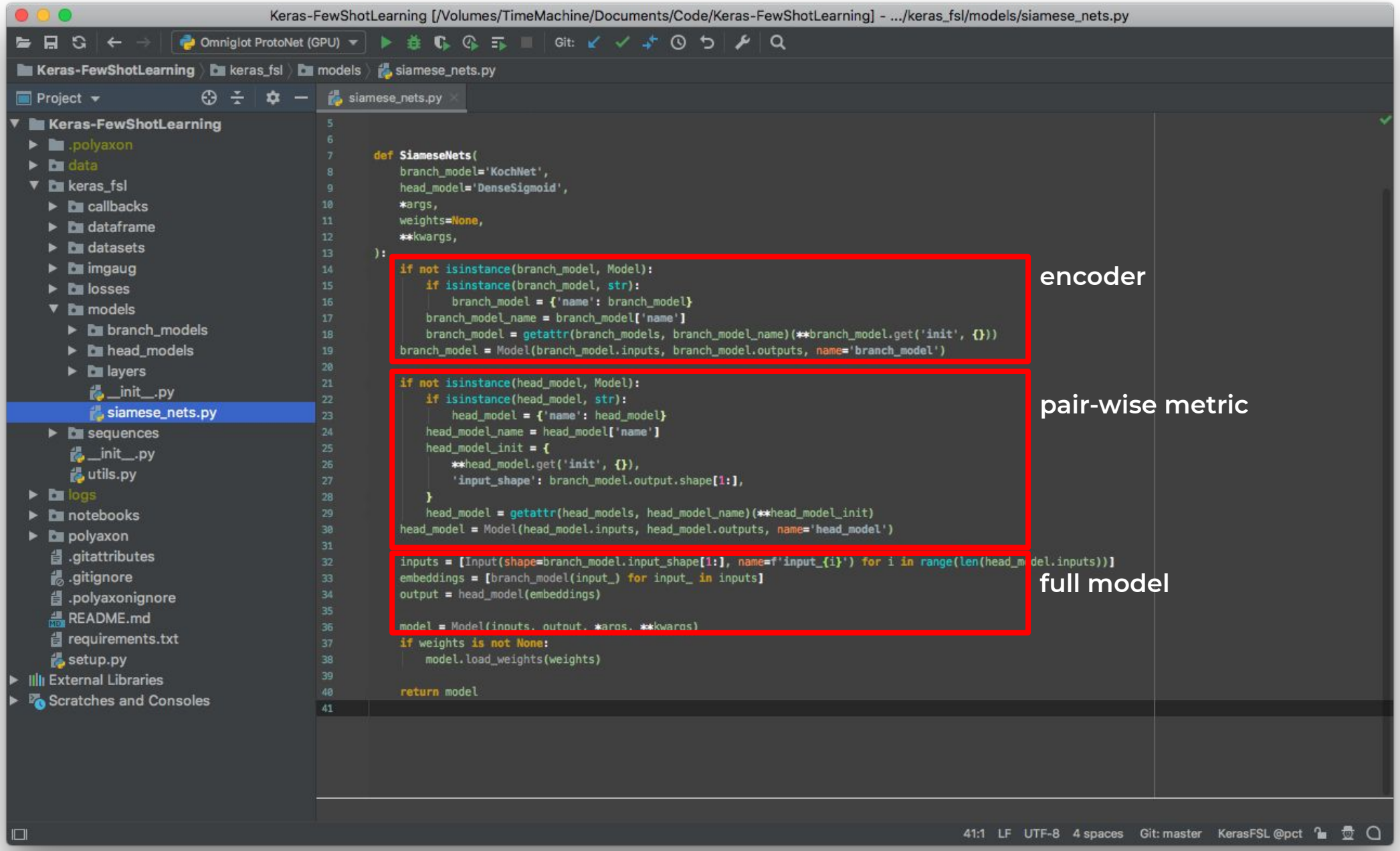
Code
keras_fsl



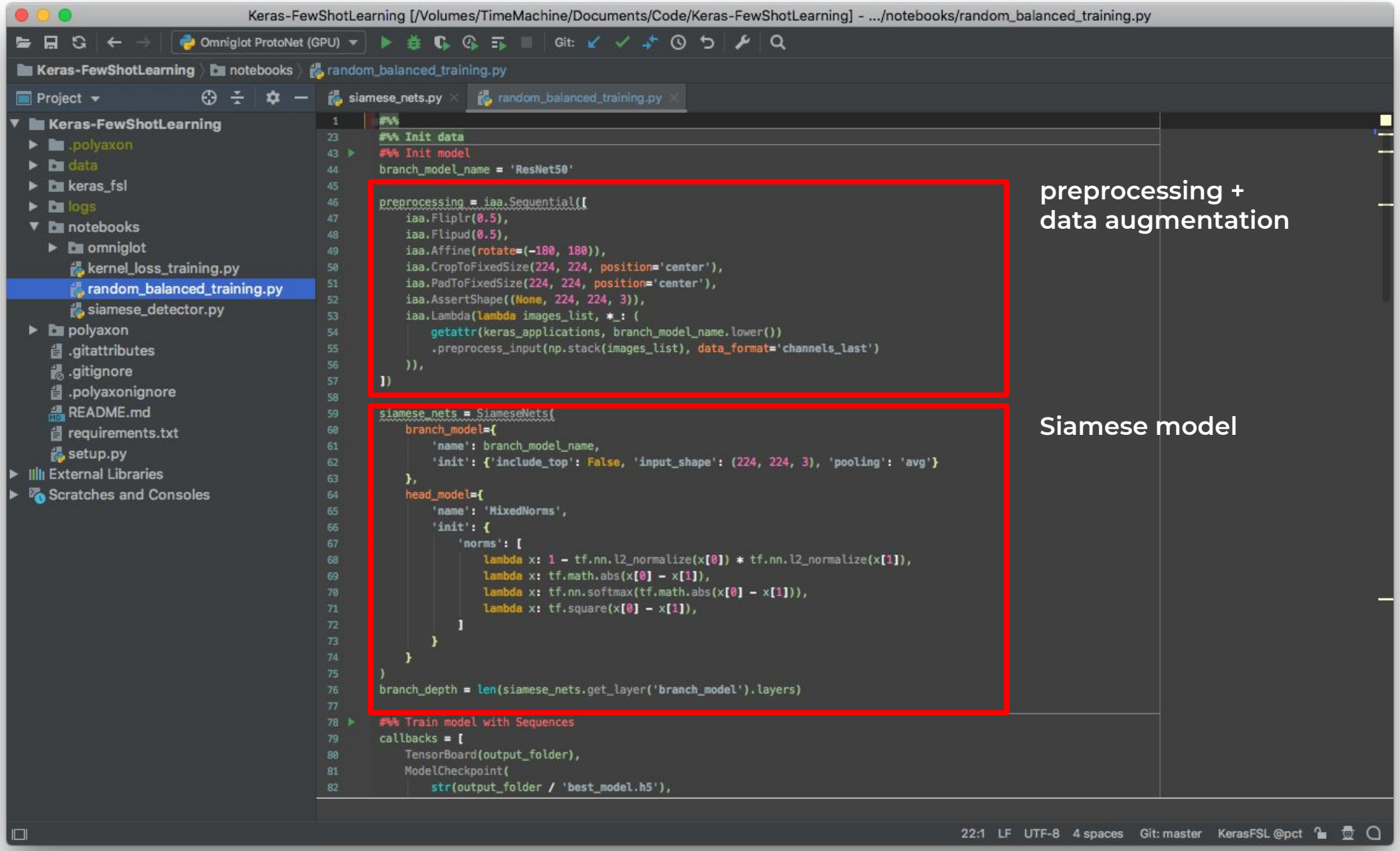
Code
keras_fsl



Code
keras_fsl



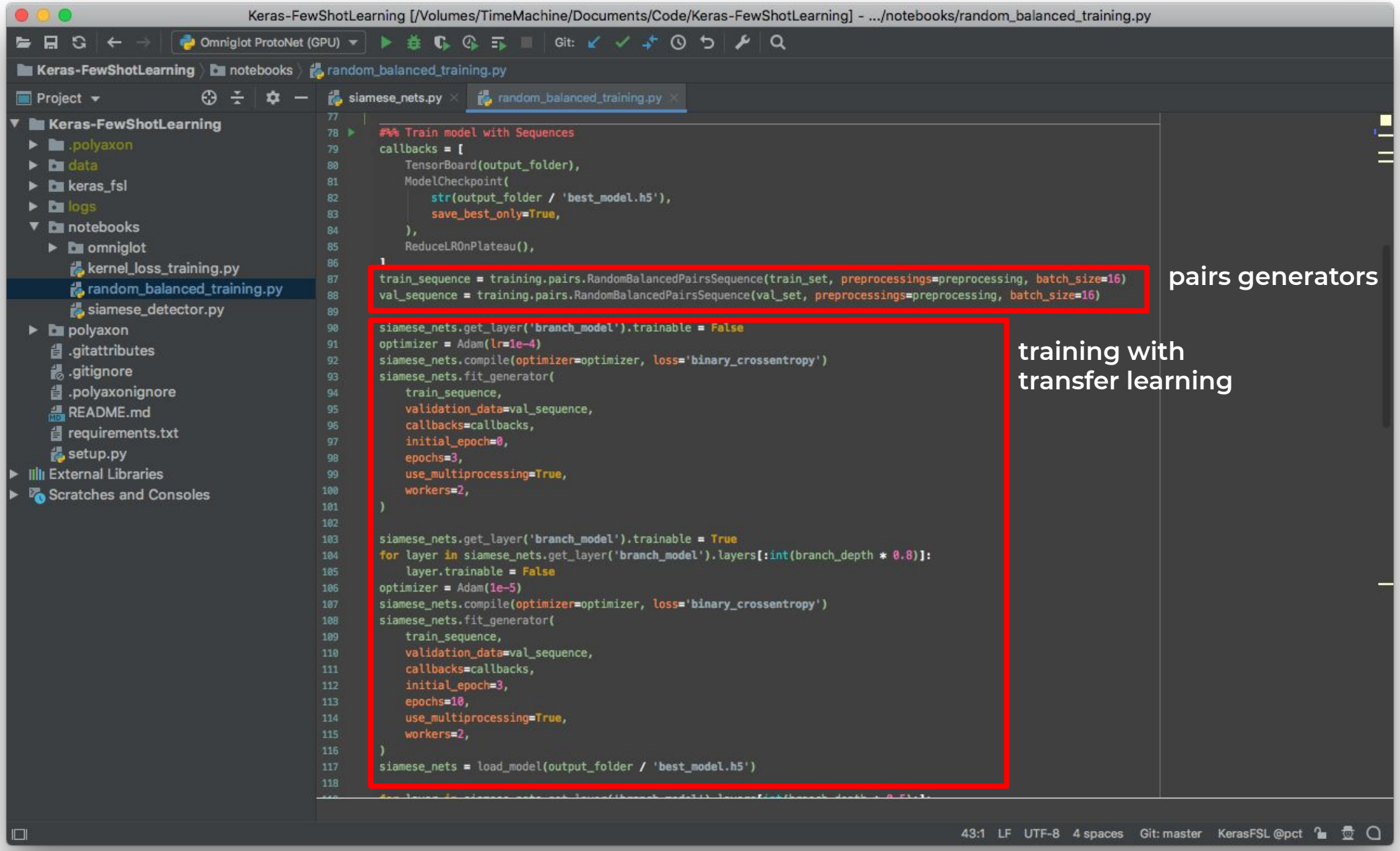
Code notebook



preprocessing + data augmentation

Siamese model

Code notebook



SOMMAIRE

Few-shot learning
CONCLUSION

Meta learning with retraining

- Best performers on benchmarks, but do benchmarks matter?
- Requires retraining whenever the support set changes

Metric learning

- One single training
- Siamese net with simple classifier (nearest neighbors) more suitable to production diversity

Siamese nets as usual classifier with specific loss

- #pairs = (batch_size)^2
- Output usual classifier with internal support set for inference

Ressources

- kaggle : https://www.kaggle.com/c/humpback-whale-identification/notebooks
- nanonets.com
- https://github.com/ClementWalter/Keras-FewShotLearning


SICARA
Merci