CONVISTA:

HS Ansbach Summer Term 2023 – Advanced AI

Pracitcal Assignments – Eurosat: Satellite Image Classification – 26.05.2023

1_ General Information

There will be marked assignments you will need to complete to pass this course.

Make sure that you are a member of the correct group in our Moodle course.

All the information regarding the groups will be available in Moodle.

Please upload your material to Moodle until July 21st, 2023:

- Python notebook(s) of the code (make sure it is reproducible)
- Optional: group presentation from June 30th, 2023
- Optional: Other explanatory documents (it may be easier for you to work on several steps in other document-types like Text-Documents, Tables, Slides, …)

We require 10 pages of material per participant (remember that you can submit code, pictures, text, …).

Please ensure to denote/mark who contributed what part in order to allow for an individual evaluation.

If you have questions regarding the project please use the forum in the Moodle course!

2_ Task Description

<u>Data:</u> "EuroSAT.zip"

The folder "2750" contains ten sub-directories ("AnnualCrop", "Forest", "HerbaceousVegetation", "Highway", "Industrial", "Pasture", "PermanentCrop", "Residential", "River" and "SeaLake") with pictures of one class each. Each of the folders contains between 2000 and 3000 ".jpg"-files. Each image is a satellite image showing the respective area of surface. Classifying pictures of this kind can be useful for many applications including agricultural or urban planning.

More on the dataset:

[1] Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. Patrick Helber, Benjamin Bischke, Andreas Dengel, Damian Borth. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2019.

[2] Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. Patrick Helber, Benjamin Bischke, Andreas Dengel. 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018.

General Task: Come up with a Deep-Learning-Model that classifies a given image into one of the ten classes. You must use at least one of the network architectures from the lectures, a Transfer Learning approach using the CNN-model VGG-16 is recommended.
The following tasks give you some idea of how to structure the project. Feel free to experiment with and submit additional ideas and steps if you want to.

Tasks:

1. Outline the Business Value of Use-Case. Outline the criteria a model would have to fulfill in order to add value. (see Phase 3 in CRISP-DM Model)
2. Describe the data set at hand in-depth (see Phase 4 in CRISP-DM Model)
    a. Describe (a subset of) pictures from each class. Plot some representation for each class and describe its characteristics.
    b. Describe the technicalities of the data set, i.e. image size, dimensions (pixels) of the data. Is the data homogeneous? Do the images have different sizes? Are (all) images quadratic?
    c. Talk about the image representation.
    d. Talk about the distribution of the classes.
    e. Describe suspicions observations, i.e. anomalous images or special images that do not seem to fit the characteristics of the other images in the class. Give some possible reasoning for the observations and assess how to deal with those.
3. Preprocess the data to make it ready for use in Machine Learning (see Phase 5 in CRISP-DM Model)
    a. Load the dataset into a Python-Notebook
    b. If necessary: Resolve the issues encountered in 2.e.

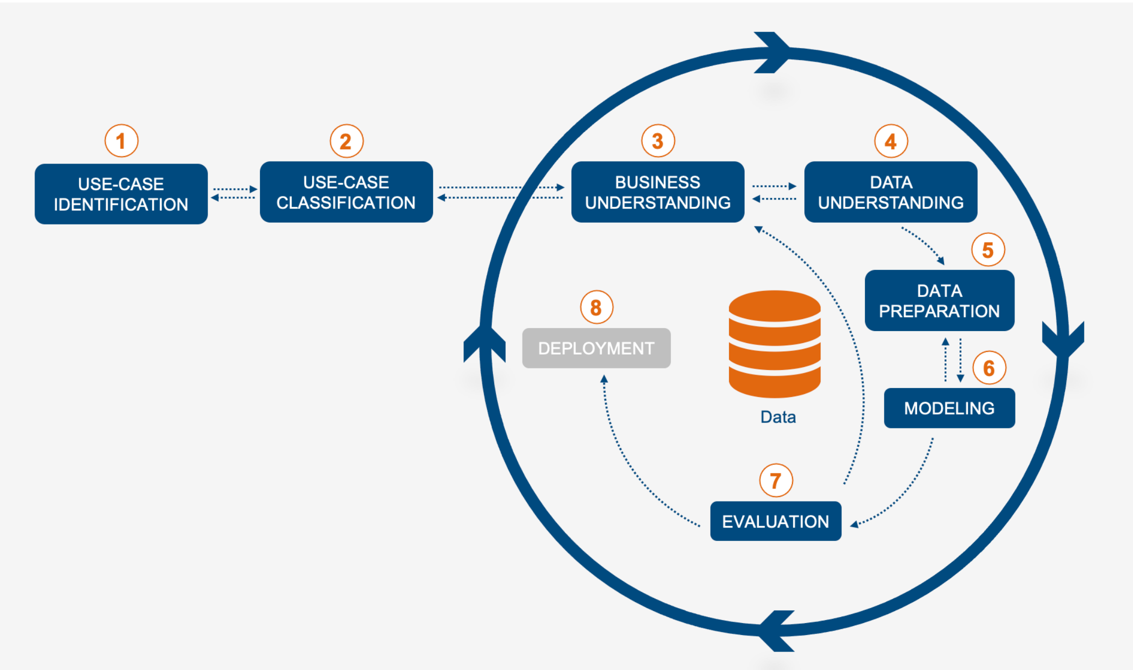
   c. Rescale images if necessary.
   d. Normalize the images if necessary.
   e. Split the data into train and test-data (possibly train, validation and test data if you need to) in a suitable way.
4. Train a DL-model for a relevant architecture encountered in the course (Recommendation: CNN) in order to classify the images from the eurosat dataset. (see Phase 6 in CRISP-DM Model).
   a. Describe why you want to use the model(s) of your choice for this specific use-case.
   b. Train a baseline Feedforward-NN model first in order to compare the performance to your main model. Use at least 2 hidden layers in your model. Use a suitable batch-size, epochs, optimizer and learning rate.
   c. Train a standard CNN model with at least 2 convolutional & max-pooling layers and at least 2 dense layers after the convolutional layers.
   d. Train both the Feedforward-NN and CNN model again with certain optimization and regularization techniques (i.e. Batch-normalization on different hidden layers, Dropout, L2-Regularization, early stopping etc.)
   e. Train a transfer-learning CNN-model using the preprocessed data from 3.
      i. Tip: You can use the implementation of the VGG-16 model (https://keras.io/api/applications/vgg/) in Keras. There are many guides on how to use VGG-16 for transfer learning in Keras.
      ii. Optional: Use a technique (grid-search, cross-validation, …) to improve the model performance
5. Evaluate the models trained in 4 on the evaluation criteria defined in 1.
   a. Use the feasible data set for the evaluation (usually your test set). Interpret the results. Use different evaluation metrics (Precision, Recall, F1-Score) on your train- and testset
   b. Plot and interpret a confusion matrix for your model(s).
   c. Compare the models <u>before</u> you retrained them with certain optimization techniques in 4: Which shows the best performance?
   d. Compare the models <u>after</u> you retrained them with certain optimization techniques in 4: Which shows the best performance? Why did the model metrics improve or worsen compared to the models before the optimization techniques?

Write a project conclusion and outlook of the next steps you would recommend. (see Phase 7+8 in CRISP-DM Model)

3_ Proposed Project Structure

It is recommended to structure the assignments using the well-known Crisp-DM model.

We have extended this model by two additional phases (1 and 2) to illustrate the challenges encountered when identifying and evaluating a potential Use-Case.



An in-depth explanation of the classical Crisp-DM (phases 3-8 in the graphic above) can be found in many places on the internet (good example: With, Hipp: CRISP-DM: Towards a Standard Process Model for Data Mining, 2000).

Let us briefly describe what you could do in the different phases of the model and how you can structure your assignment:

1) *Use-Case Identification:* You will need to think about potential Use-Cases in your domain of Smart Energy Systems: Where do we have (a lot of) data? What are interesting problems in our field that one could evaluate in a data-driven way.

2) *Use-Case Classification:* Evaluate the Use-Cases identified in 1) and come up with the most promising one. Possible criteria: Amount, quality and availability of data. Business value in solving the Use-Case. After this phase you would have identified one Use-Case for applying DL. In the following points

3) *Business Understanding:* Understand and outline the objectives and requirements of the project. Assess the current situation and how using Machine Learning/Deep Learning can be of benefit. Take risks into account as well. Define your goals and what you wish to achieve. This has to be taken into account in the "Evaluation" phase at the end of the cycle. The output is a project plan where you select technologies you want to use (which forms of DL models are to be used) and define a plan for the ensu-ing phases.

4) *Data Understanding:* Collect the data set. Describe and explore the data (what are the explanatory variables, for supervised learning: what is the target variable (label)). Possible approach: Use visualization tools, correlation analysis, PCA, describe do-mains of the covariates, … Verify data quality and document issues like anomalous, unplausible or erroneous data. You should be able to show that you understand the data and its challenges.

5) *Data Preparation:* Select the data you want to use. Clean the data is very important ("garbage in - garbage out"). Derive other interesting variables (feature engineering). You possibly have to integrate data from different sources. Choose a feasible ma-chine-readable format for the covariates. Look into standardizing the data which is generally a good idea. Split the data into train, validation and test sets.

6) *Modeling:* Now you can start actually training DL models. Sometimes it is a good idea to first establish a baseline model with more simplistic approaches from classical Ma-chine Learning. Select a DL-algorithm (or multiple) that you want to try. Then build your model. This is just one of the many steps, often it is a rather simple one. You will typi-cally revisit this phase after the "Evaluation"-phase: Try different hyperparameter combinations. Maybe you implement a grid-search algorithm in order to find better hy-perparameters in an analytical way.

7) *Evaluation:* Evaluate you model(s) with the previously defined criteria (Business Un-derstanding). This is a good point to take a step back and look at the process: Can you improve on any of the previous steps? Determine the next steps: Is the perfor-mance good enough? If not: Maybe you revisit the previous phases with your new and improved understanding of the subject. At the end of this phase you should be able to come up with a wholistic evaluation: Which is the most useful model? What can you achieve with the model? What are its possible limitations?

8) *Deployment:* Outline the next steps needed to make the model "production ready"? You should produce some kind of final report of what you did during the previous phases and review the project.