2AMM10 Assignment 1:
Analyzing Persian Carpets with ML

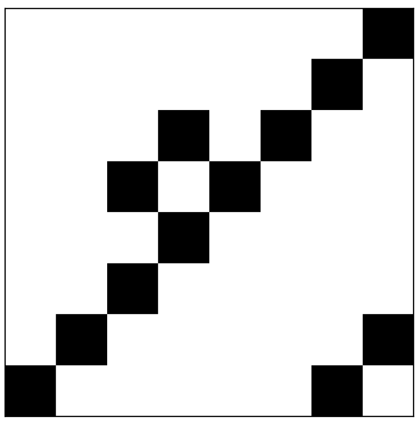
1 Introduction

Persian carpets are heavy textiles made for both practical and symbolic purposes in Iran. Carpet weaving is an essential part of Persian culture, and has been a part of Persian history since the 5th century BC. In addition, carpets made in different regions of Persia are characterized by different weaving techniques, materials, colors, and *patterns*.

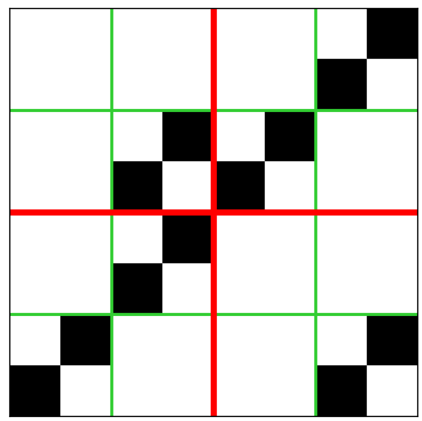
Figure 1: Example of a Persian carpet

In terms of their texture, typically multiple levels of patterns are present (Fig. 1). On the highest level, a grid-like structure is present, where each grid cell contains further sub-patterns. At the lowest level (or smallest scale) patterns are represented by a *unit cell* and can occur in multiple locations on the carpet. A group of *unit cells* together comprise a *super cell*, which can also occur in multiple locations on the carpet. Finally, on the highest level of the carpet, it has a pattern defined by the *super cells* (global pattern).

To better understand the structure of these carpets, we look at a simplified example given in Fig. 2a. This carpet can then be further split into (sub)patterns (Fig. 2b), where green lines denote *unit cell* borders, and red lines denote *super cell* borders. If we take a closer look, we can notice there are only 2 types of *unit cells* (Fig. 3): one has black pixels on the top right and bottom left (U_A) and one consists of only white pixels (U_B). Similarly, if we look at the arrangement of *unit cells* inside a *super cell*, we can notice two types of *super cell* (Fig. 4). In S_I, *unit cell* U_A appears on the top right and bottom left location (A), and *unit cell* U_B appears on the main diagonal (B). For S_{II}, U_A appears in the bottom right corner (A), while the rest of the *super cell* is filled with U_B (B).



(a) Example carpet



(b) With Unit/Super cell borders

Figure 2: Carpet pattern with and without unit/super cell borders

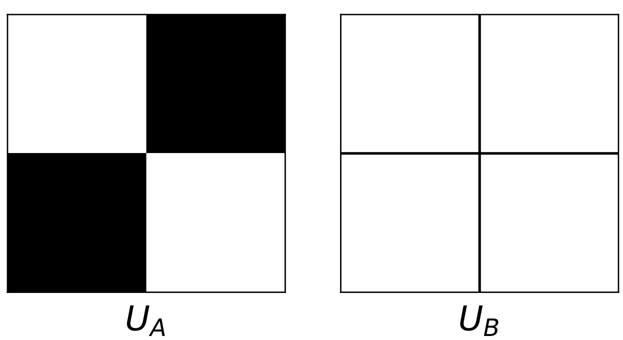


Figure 3: Two different unit cells

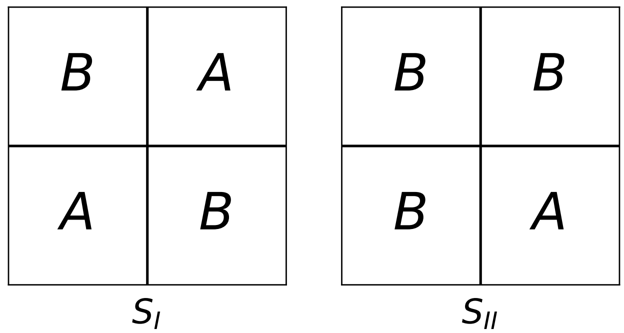


Figure 4: Two different super cells

The class of a carpet is determined by its global pattern. That means that the arrangement of *super cells* determines the carpet class. For example in Fig. 5 a global pattern is given. This global pattern can be specified as: the *super cell* at position (1, 1) is equal to the *super cell* at position (2, 2) and the *super cell* at position (1, 2) is equal to the *super cell* at position (2, 1).

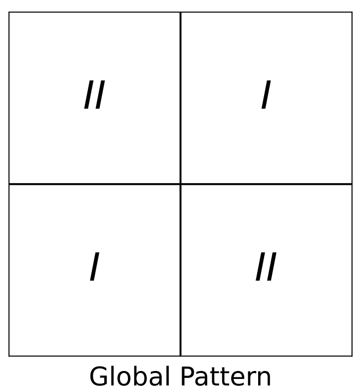


Figure 5: Global Pattern of the example

Notice here that the *super cells* at specific locations are not specified in the pattern. Therefore, the global pattern can be matched by many different *super cell* arrangements. An example of this is given in Fig. 6.

Finally, in the given dataset there are many more than 2 types of *unit cells* and 2 types *super cells*.

In this assignment, your task will be to develop Deep Learning models to classify

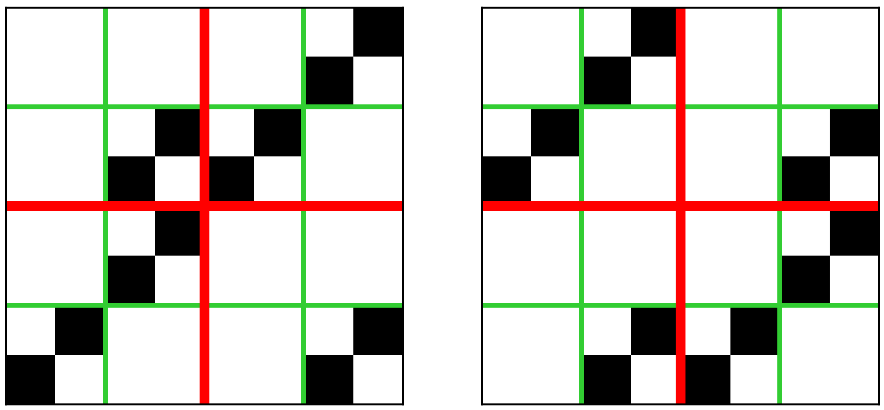


Figure 6: Two equal global patterns.

and compare different kinds of Persian carpets. In both tasks, you should keep in mind the structure and symmetries of the data, and use this to **make your model as efficient as possible in terms of parameters and operations**.

2 Assignment Description

As mentioned previously, we will be looking at Persian carpets. More specifically, we will be analyzing carpets from various (fictional) Persian colonies, which were founded by Khan Vlahid Mankoushi in the 16th century. Each of these colonies has their own distinct carpet patterns, which can be used to determine where a carpet is from. Due to the age of the carpets, many carpets started losing their color or got damaged. In some cases, these carpets were patched up using pieces of different carpets or random pieces of cloth. As such, rule-based methods for analyzing these carpets are not sufficient and Deep Learning methods are needed.

Next to being a colonizer, the Khan was also a big fan of mathematics. As such, he made sure that all the carpets produced in his cities follow the same patterns, placed on grids of the same sizes.

Each unit cell $\mathbf{u}$ contains a 4 by 4 grid, on which the lowest level of patterns occur. The values on this grid are binary. Mathematically, we can define this grid as: $\mathbf{u} = \{u_{ij} \mid u_{ij} \in \{0, 1\} \wedge i, j \in \{1, 2, 3, 4\}\}$, where u_{ij} represents the color of the smallest tile. We denote the set of all existing *unit cells* with U. Since the *unit cells* are the first part of the carpet which is stitched together, mistakes can occur as a result of which the patch can be rotated. In practice, $\mathbf{u}$ can be rotated over $k \cdot 90°$, where $k \in \{0, 1, 2, 3\}$. This rotational symmetry is isomorphic to the cyclic group of order 4 (C_4).

The *unit cells* are contained in a 3 by 3 *super cell* $\mathbf{s}$, which is defined as $\mathbf{s} = \{s_{ij} \mid s_{ij} \in U \land i, j \in \{1, 2, 3\}\}$. We denote the set of all *super cells* with S. As the *super cell*s are larger patches of carpet, the weavers were more careful and never made mistakes regarding the rotation of these pieces of carpet.

Finally, the carpet's global pattern is defined on a grid of 8 by 5 *super cell*s, defined as $G = \{g_{ij} \mid g_{ij} \in S \land i \in \{1, 2, ..., 8\} \land j \in \{1, 2, 3, 4, 5\}\}$. In total, the Khan defined 18 different *unit cells* ($|U| = 18$) to be used on his carpets, as well as 12 different *super cells* ($|S| = 12$). Due to the complexity of the carpets, there is a chance of having the wrong tile at each level.

In [1] the authors present a method that may help with ideas about how to incorporate related data symmetries in the model architecture. In the file symconv.py, some of the operations are implemented: Slice, Pool, and Conv2d with a kernel that can rotate and be reflected. Contrary to [1], we add an extra dimension to represent the symmetries instead of including these in the batch dimension. As such, rather than rotating the entire grid, we only need to rotate the kernel.

While different types of models might obtain good performance on both tasks, it's important that your models account for **all** symmetries described to be in the data.

2.1 Task 1: Carpet Origin Prediction

While the Khan founded various cities throughout his life, three cities have grown a particular big carpet-weaving industry: Convolushahr, Transformabad, and Reinforciya. As a result, we have many examples of carpets produced in these cities. Since the carpets from these cities spread around the world, many of these carpets are found every day. To quickly determine which city the carpet was made in, we want to develop a model.

The goal of this task is to develop a method that can classify in which city a carpet was made, based on the pattern of the carpet. The training dataset consists of 7500 carpets, with their associated labels, the validation dataset consists of 2000 carpets and their labels and the testing dataset consists of 500 carpets and their labels. Each of these carpets follows the structure as described earlier.

2.2 Task 2: Carpet Matching

As the Khan was an avid explorer, he also founded many smaller towns and settlements, which did not have a carpet industry. As a result, carpets from these places were often handmade, and only a few carpets from each village survived. For this collection of carpets, it is known from which village they originate. Recently, an archaeologist found a cache containing carpets from various forgotten villages founded by the Khan. Carpet researchers have found 4 similar (different) carpets for each newly discovered carpet. However, they were not able to determine with which of these carpets the pattern is shared, and are therefore still unsure from which village the carpets came from.

In this task, you need to create a model for carpet matching. The carpet matching task is defined as follows: Given a query carpet x_q, and a set of 4 candidate carpets $\{x_1, x_2, x_3, x_4\}$, your solution should match the query carpet with one of the candidate carpets based on their origin. As training data, you will be given a dataset containing 15000 carpets from 200 different settlements, which you need to use to train a suitable solution. To evaluate your solution a test dataset is provided consisting of 300 queries (each query has one x_q carpet and four candidate carpets). However, the carpets in the test dataset come from a different region of the country and their origin does not match any of the 200 settlements given in the training data.

3 Deliverables

The submission of this assignment is done in ANS. There are two deliverables:

- Implementation of the solution
- Report describing your approach

For the implementation, a skeleton Jupyter Notebook with functions to load the data is provided, along with 2 Python files for visualizing the data and implementing convolutions with rotational symmetry. Please develop your solution in the skeleton Jupyter Notebook and upload that to ANS.
Specifically, you need to upload the Jupyter Notebook code (1) and a PDF version (2) with the execution output. The maximum upload size is 25MB. So, you may need to clean up the code Notebook from outputs before uploading. You can generate the PDF in Google Colab by going to "File → Print → Save as PDF". Please generate the PDF after running all cells of your solution.
The report is submitted as a digital group test in ANS. Rather than an unstructured document the report is formatted in a number of questions that you need to answer to describe your solution and understanding of the assignment. Please see ANS for the details.

References

[1] Dieleman, S., De Fauw, J., Kavukcuoglu, K.: Exploiting cyclic symmetry in convolutional neural networks. In: International conference on machine learning. pp. 1889–1898. PMLR (2016)