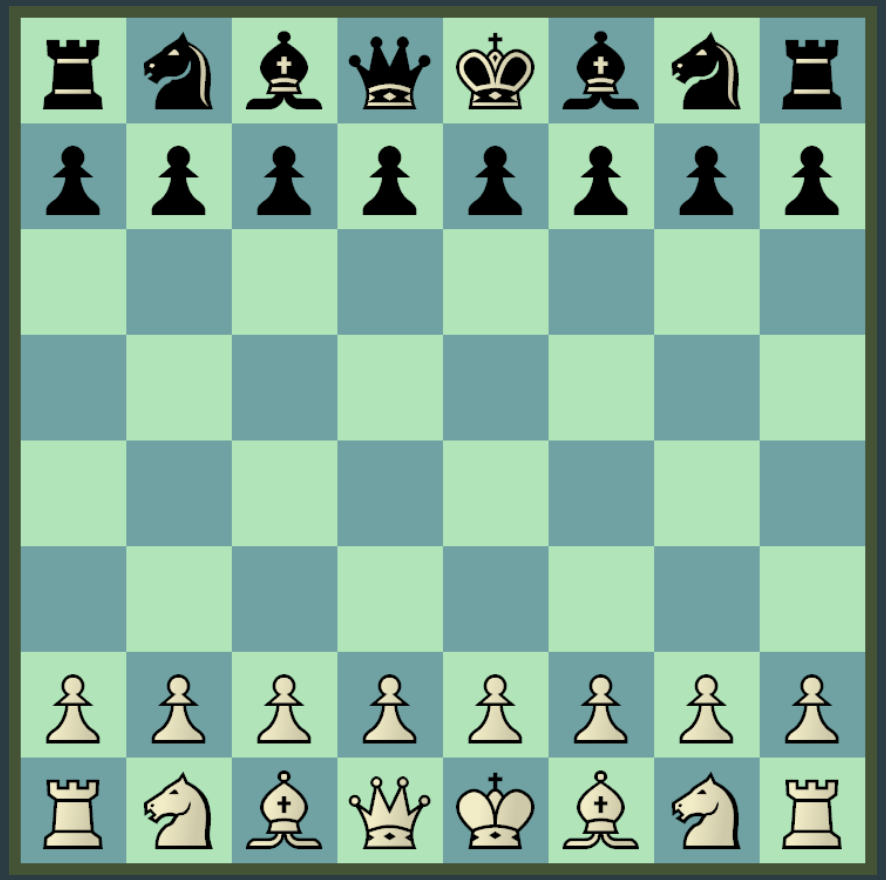
Chessboard diagram identifier



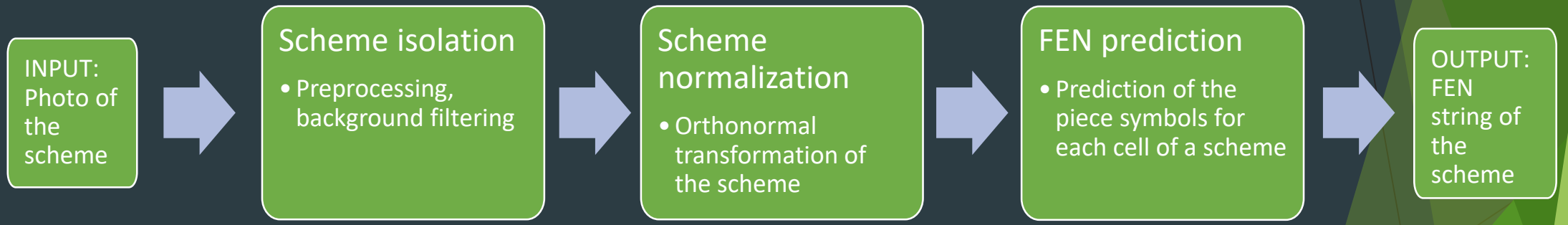
Elaborazione delle Immagini 17/18

Nassim Habbash (808292)

Objective of the project:

- From a photo of a chess scheme, predict its board configuration in the form of a FEN string.

Generalized pipeline:

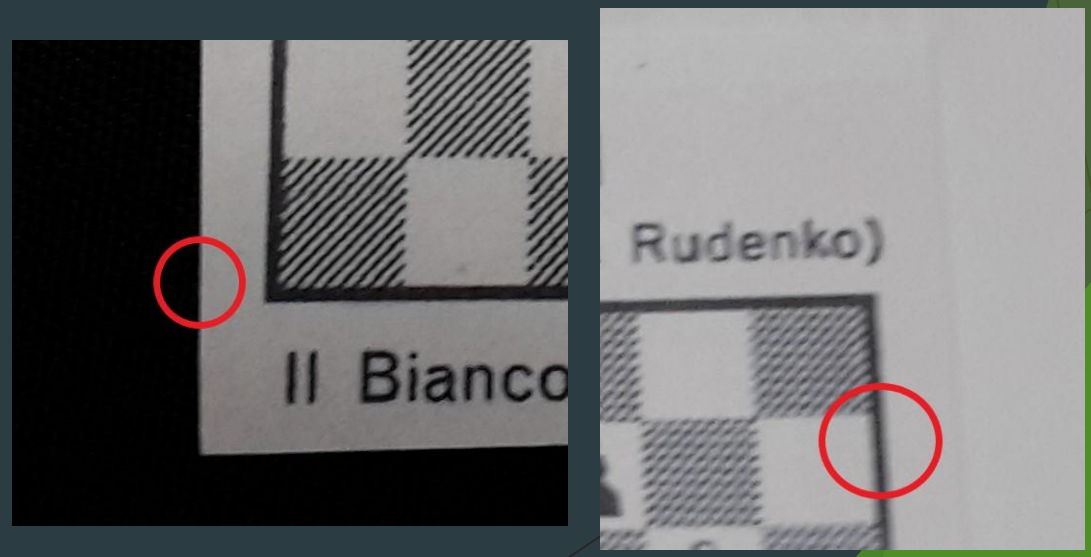


Scheme isolation:

Scheme isolation:

- The photos present different aquisition conditions
  - Different lighting
  - Hanging shadows
  - Distance of the scheme
  - Background clutter
  - Angle
  - Focus

- Chosen characteristics to isolate the schemes:
  - Solid black edge between the schemes and the background



Scheme isolation pipeline:

Preprocessing

- Filtering (Gaussian, Top Hat, …)

Edge detection

- Canny

Morphological operations

Blob Filtering

- Biggest blob is the scheme

Considerations:

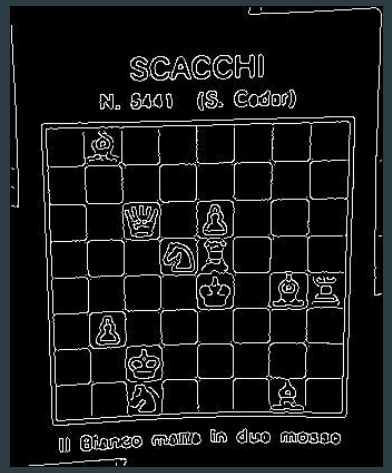
Assumptions:

- The scheme is always the biggest object delimited by a solid square edge
- It is relatively close to the camera
- The square edges are linked

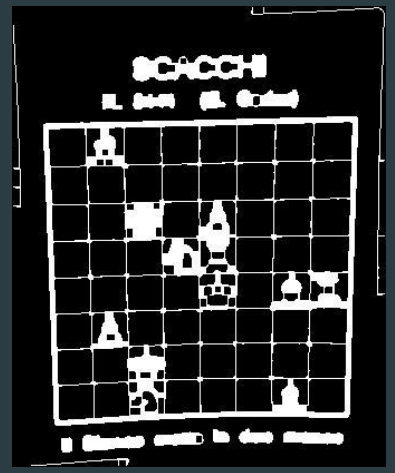
Procedure:

- Empirical and parameter-driven
- Find the issue => Find the solution to the issue => Find the parameter to make such solution work

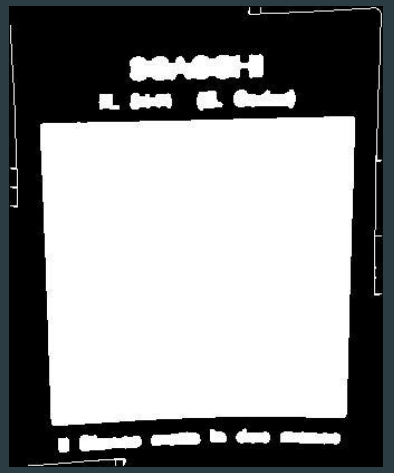
Procedure (Morphological operations step):



- Needs a solid edge => Test the right parameter to morphologically close the image



- Need to fill closed-edge areas => Flood-fill operation (imfill)

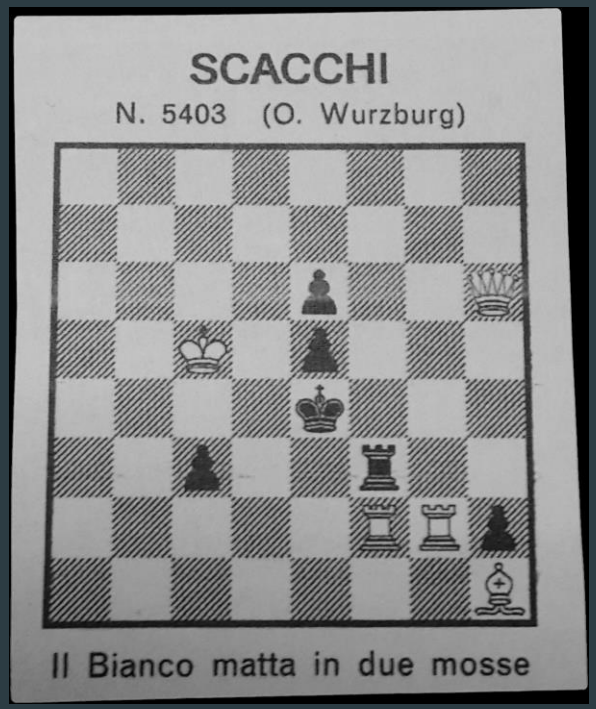


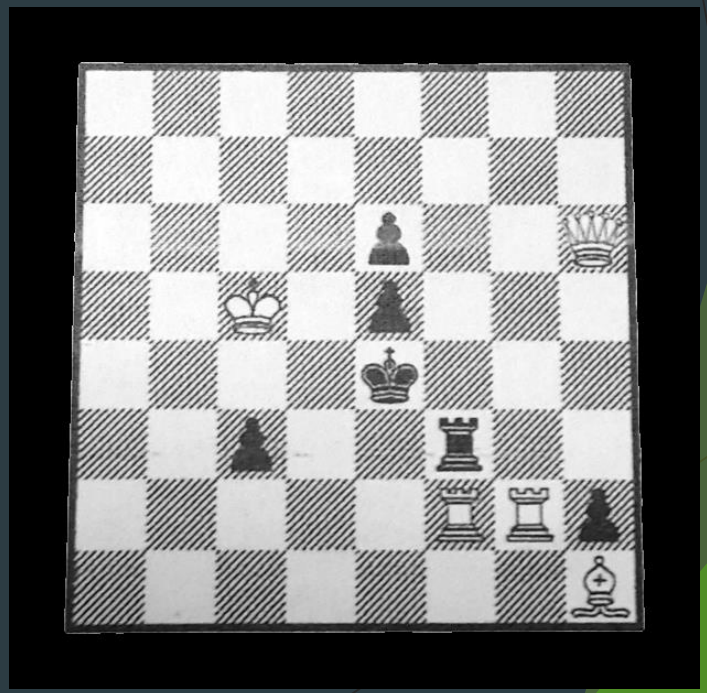
- Need the biggest blob => Area filtering

- Found the correct mask of the scheme

Approach details:

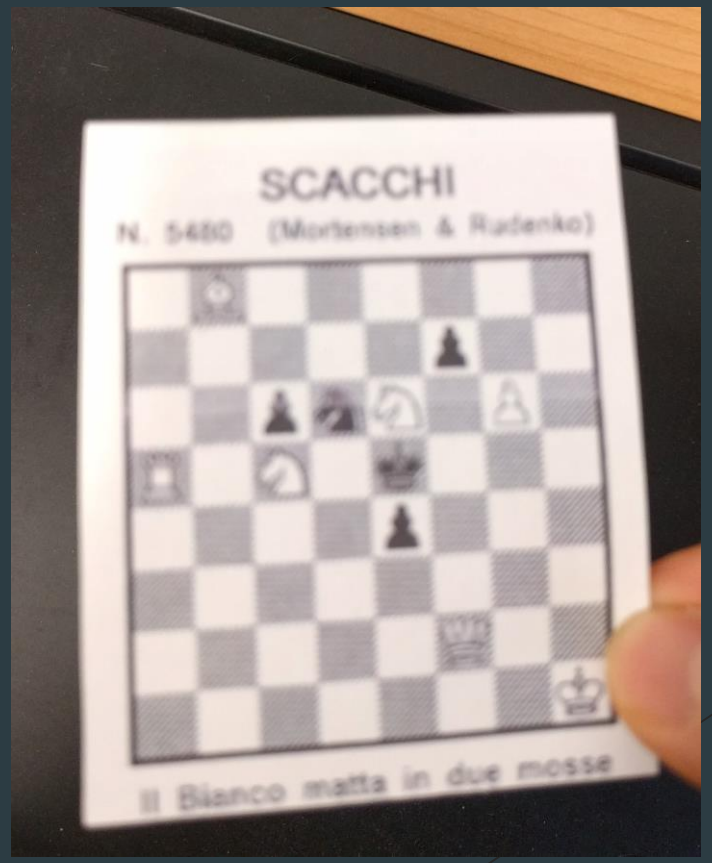
- For some photos, it is necessary to go through the pipeline two times with different parameters to remove the outer white portion of the paper (double the time of processing for each image)



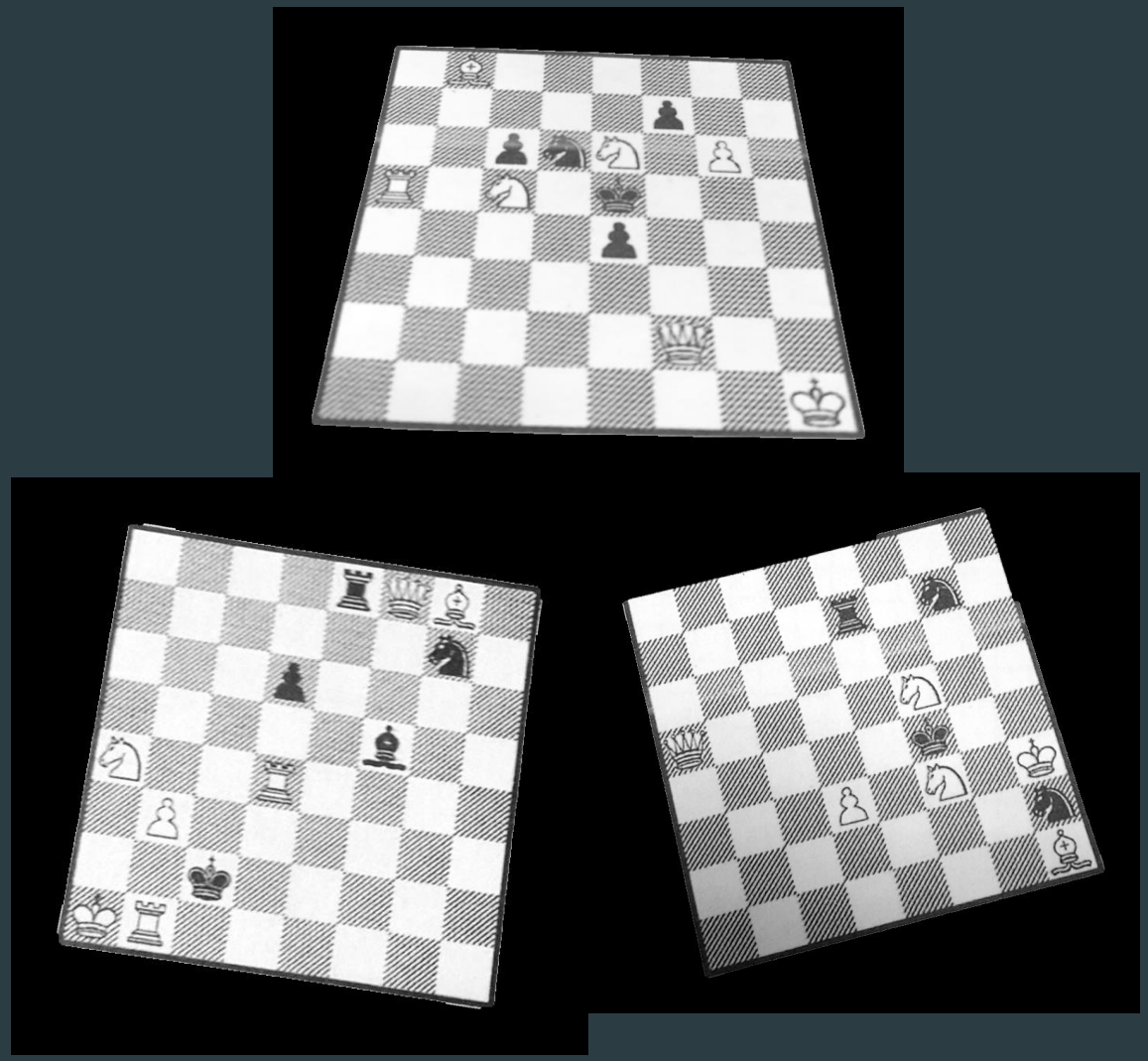


Approach issues:

- Schemes with no solid contiguous edges aren't recognized (018.jpg and 040.jpg)



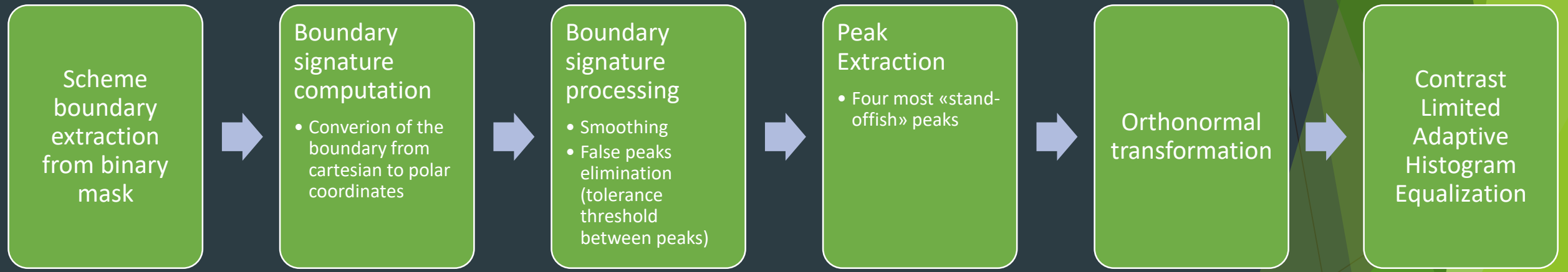
Scheme normalization:



Considerations:

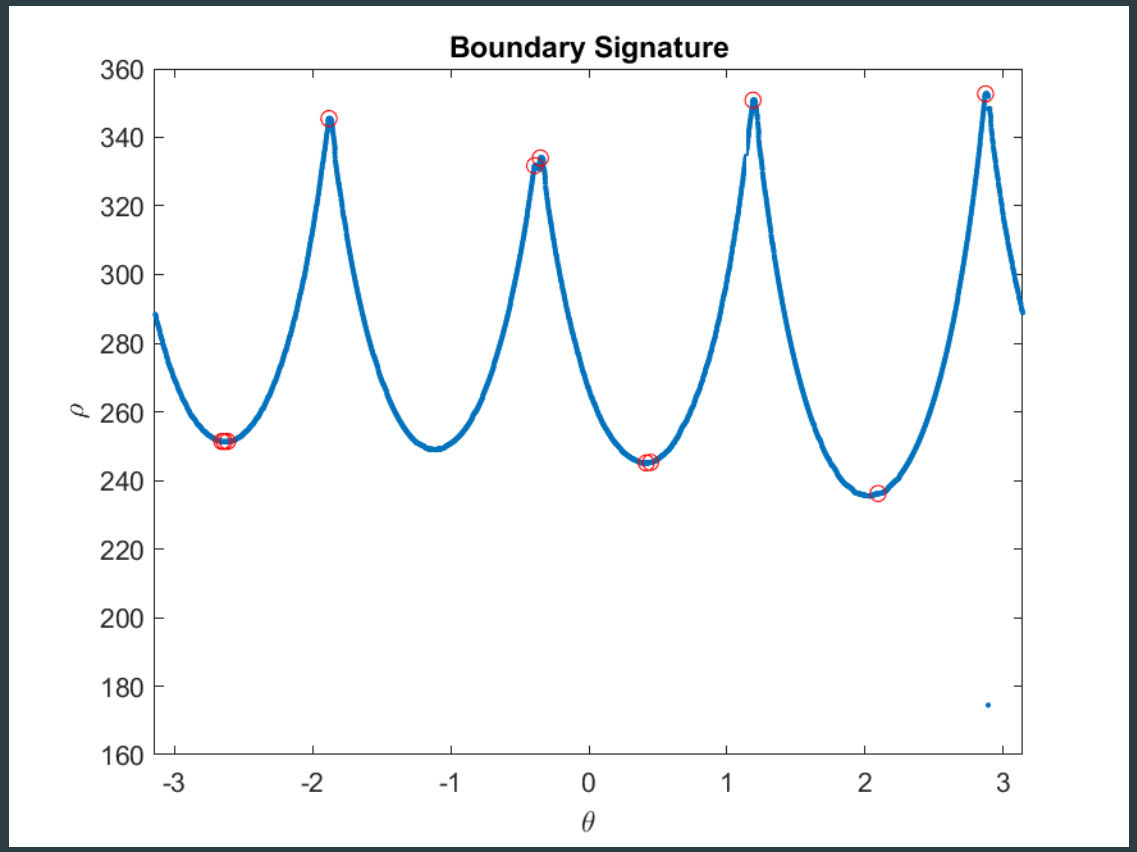
- Each scheme has a different corner location
- Locate the corners from the boundary of the scheme and apply an orthonormal transform

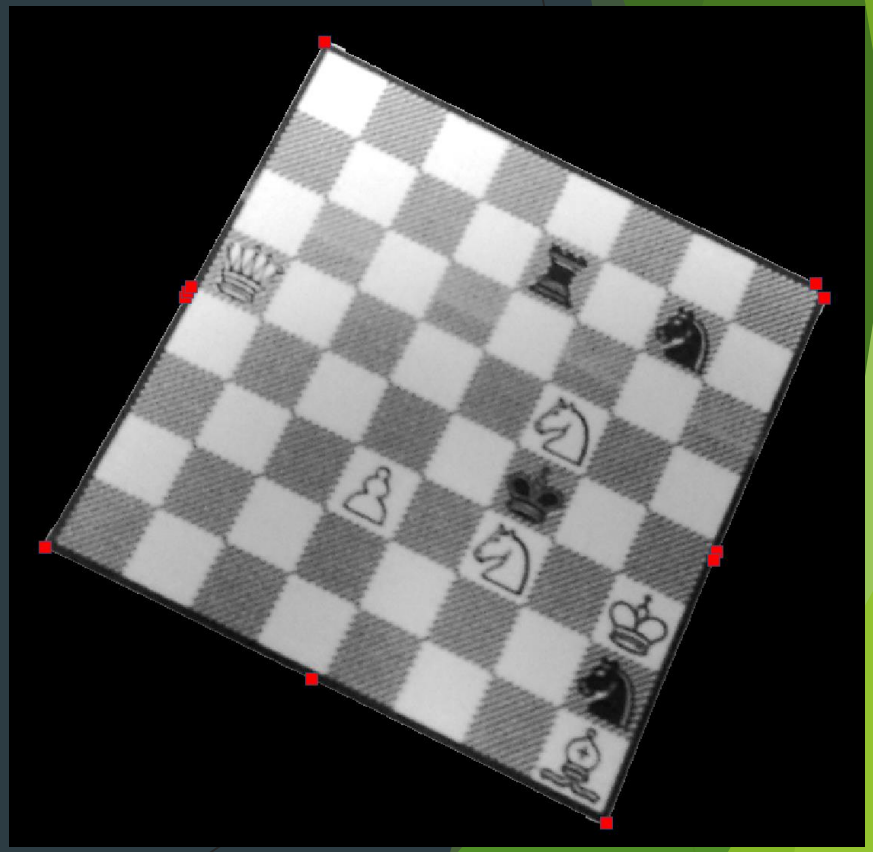
Scheme normalization pipeline:



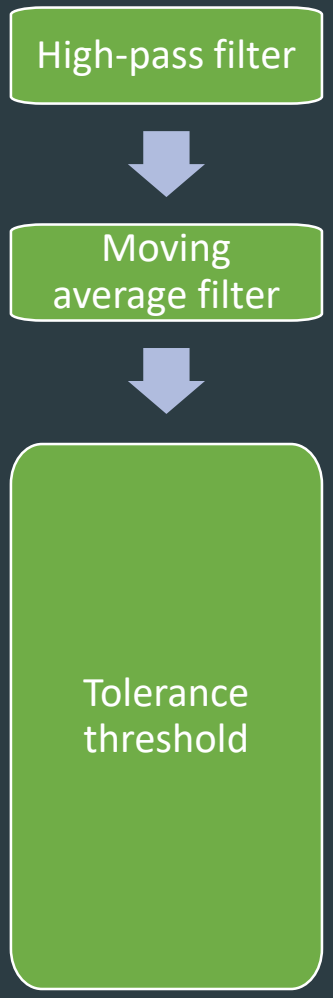
Approach – Peak Extraction:

- The boundary signatures aren't smooth and there are multiple local maxima detected. Some further processing is needed to detect the four real corners (peaks)





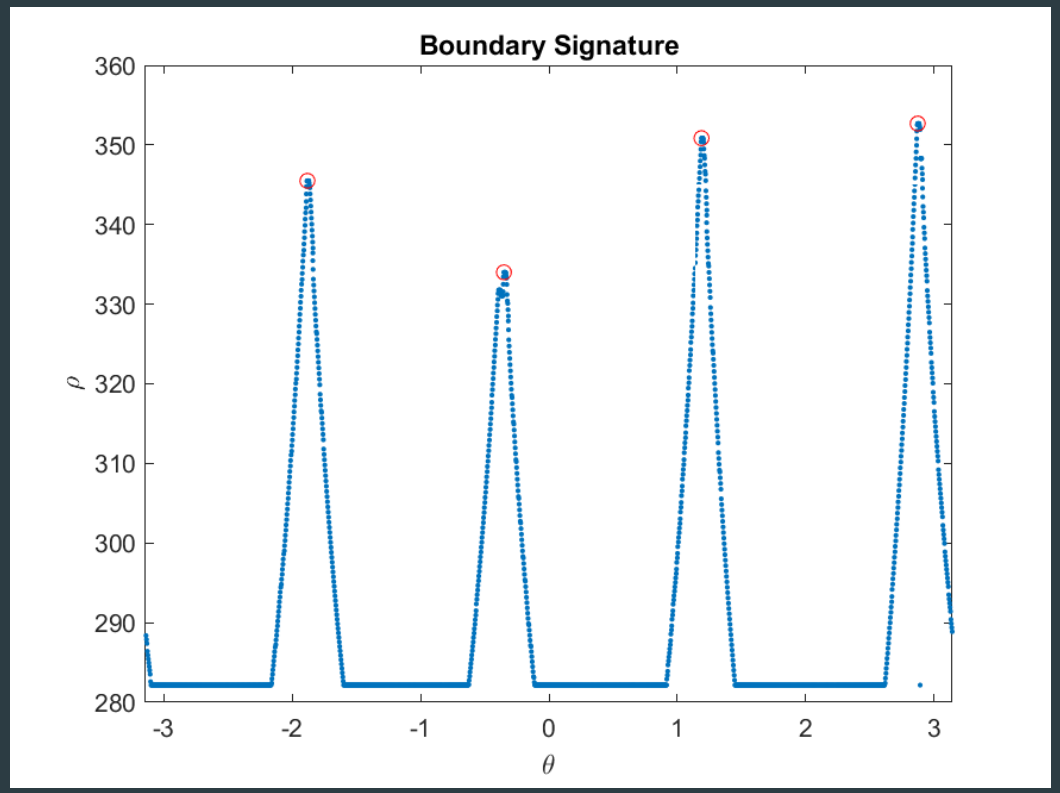
Smoothing and tolerance thresholding:

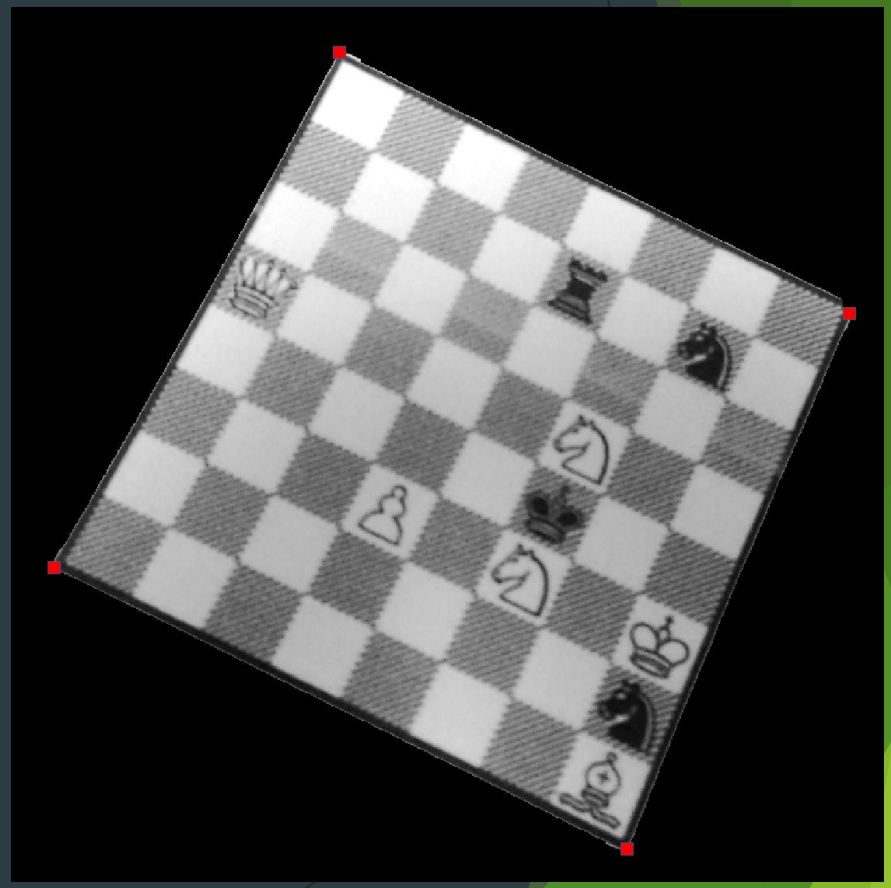


- For each couple of peaks check if their distance is within a set of tolerances. Based on which interval they fit in, they're classified as true corners or as false corners.

Approach – Peak Extraction:

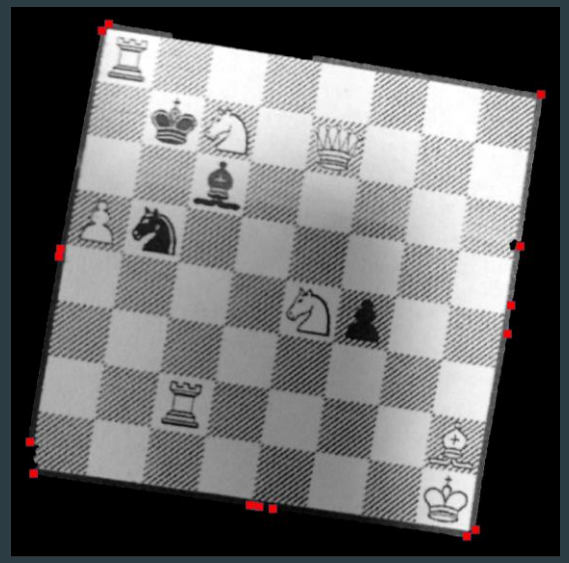
- Peaks detected after smoothing, and tolerance thresholding between peaks



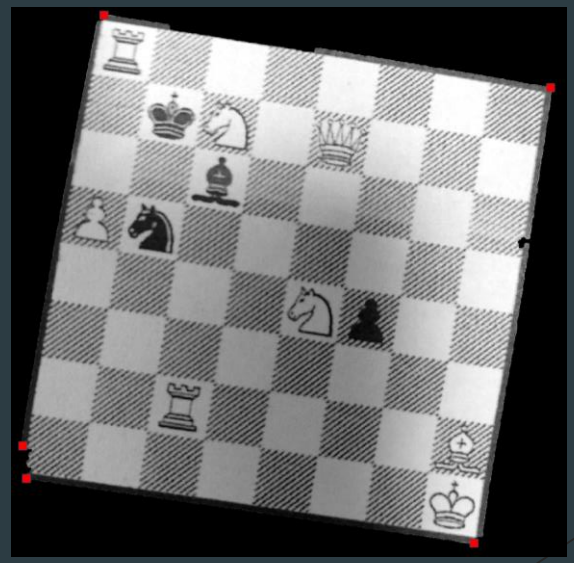


Approach issues – Peak Extraction:

- The algorithm doesn't guarantee 100% the extraction of 4 corners – it can be refined.
  - Refine the boundary signature peak detection algorithm
  - Refine the scheme isolation pipeline to output smoother edges



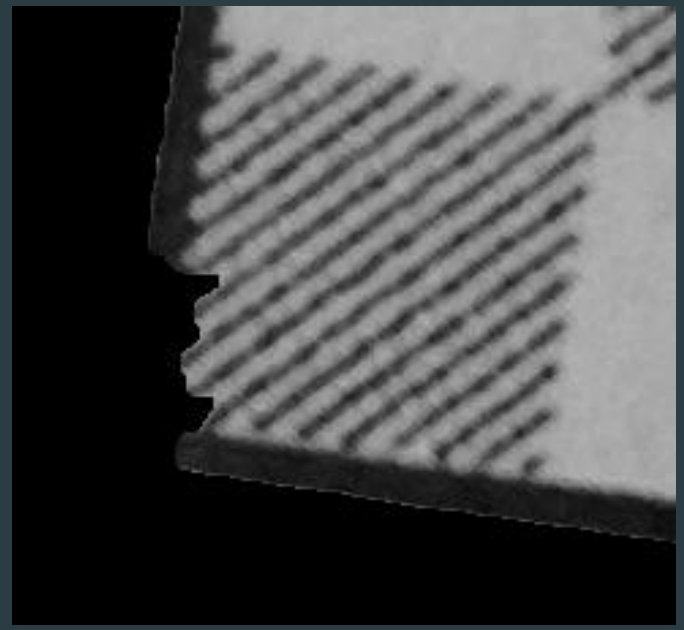
Before processing

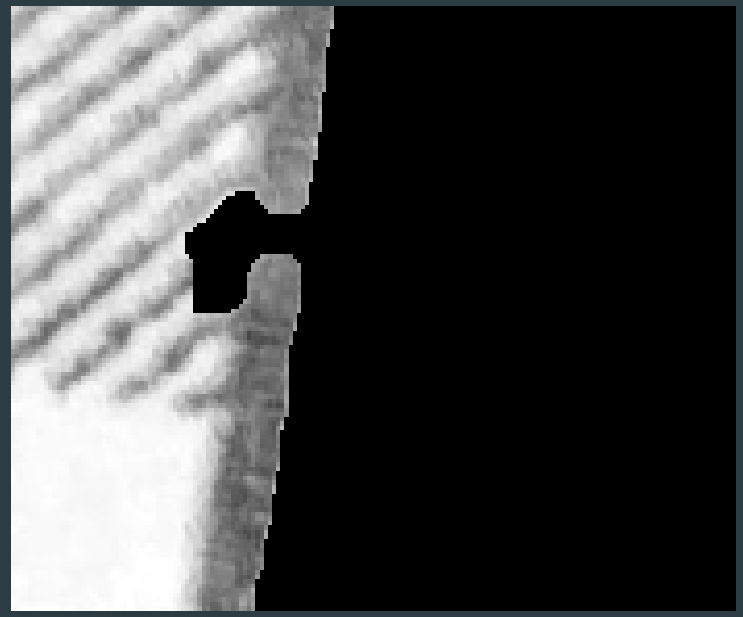


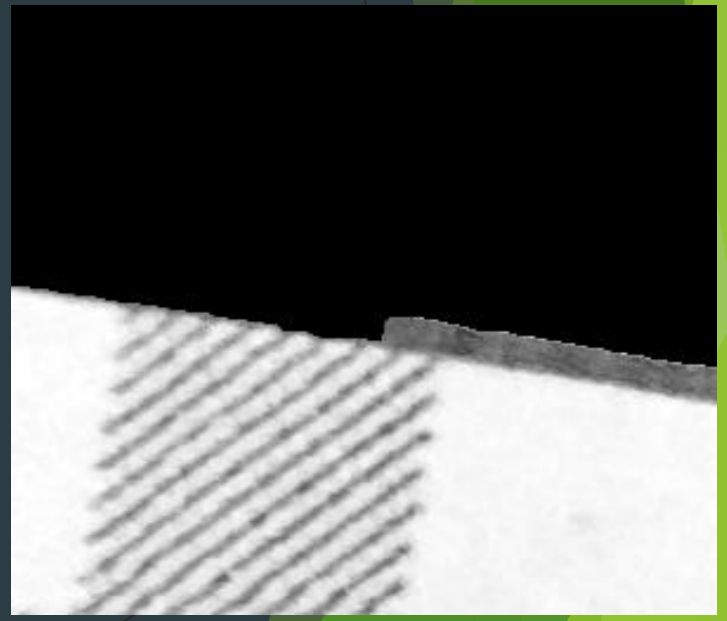
After processing: failed peak extraction

Approach issues – Peak Extraction:

- Troublesome areas







Classification:

- Train a classifier to identify a piece on a cell – from an image output its label

Black pawn (p)

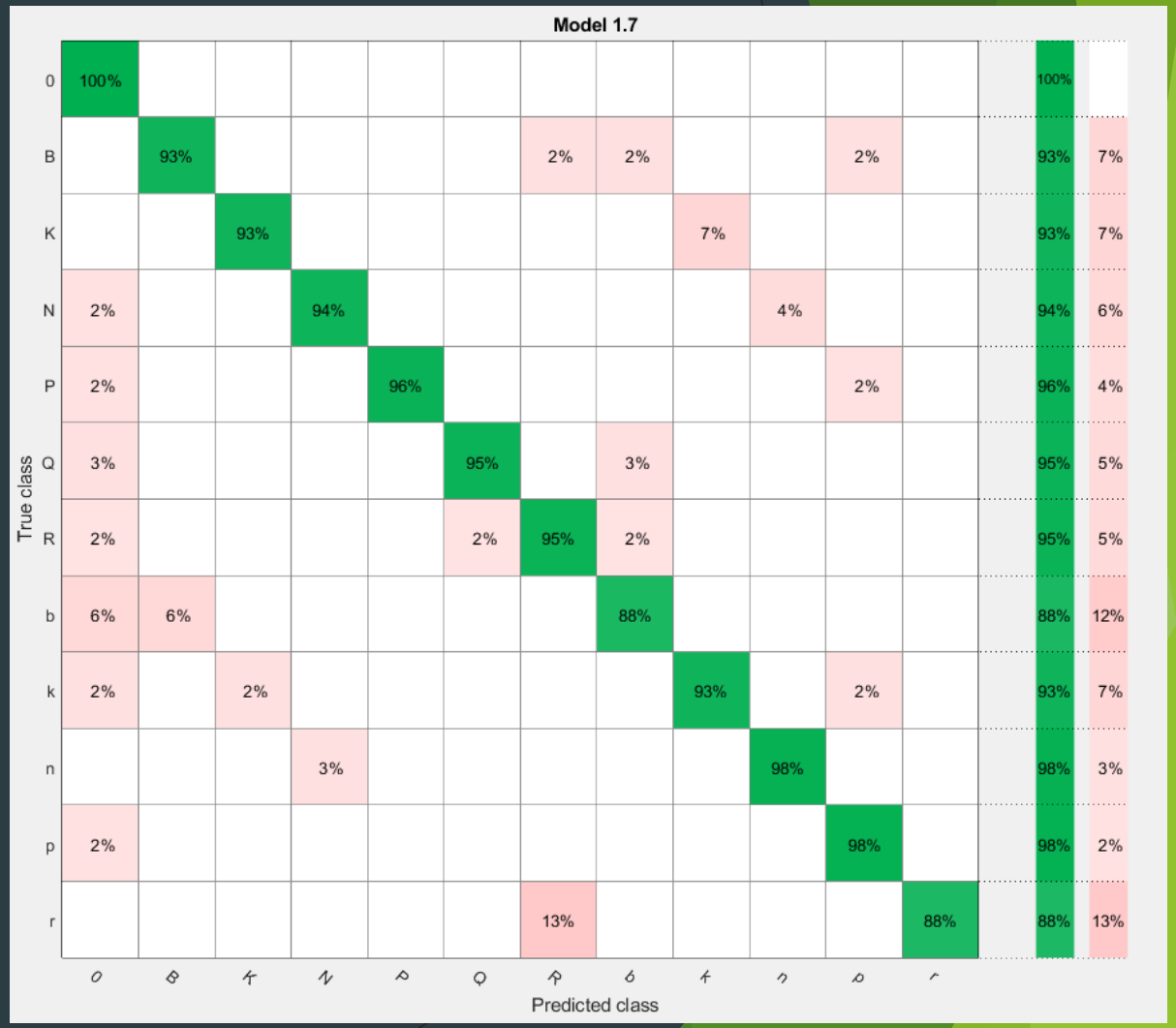
White bishop (B)

Classification:

- 40 schemes (split in 8x8 cells) are used for training/testing the classifier. 5 are used as final real case tests.
- The chosen features are:
  - Local binary pattern (LBP)
  - Histogram of oriented gradients (HOG)
  - Speeded up robust features (SURF)
  - Average
  - Variance

Classification – Confusion matrix:

- Model: Quadratic SVM
- 10-fold Cross Validation scheme
- Overall accuracy: 99%

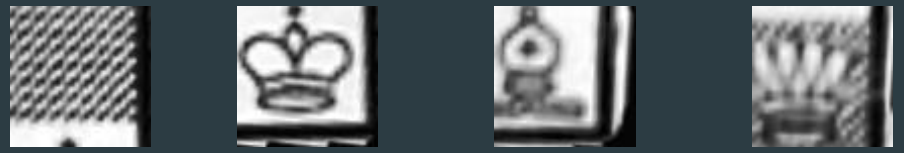


Considerations:

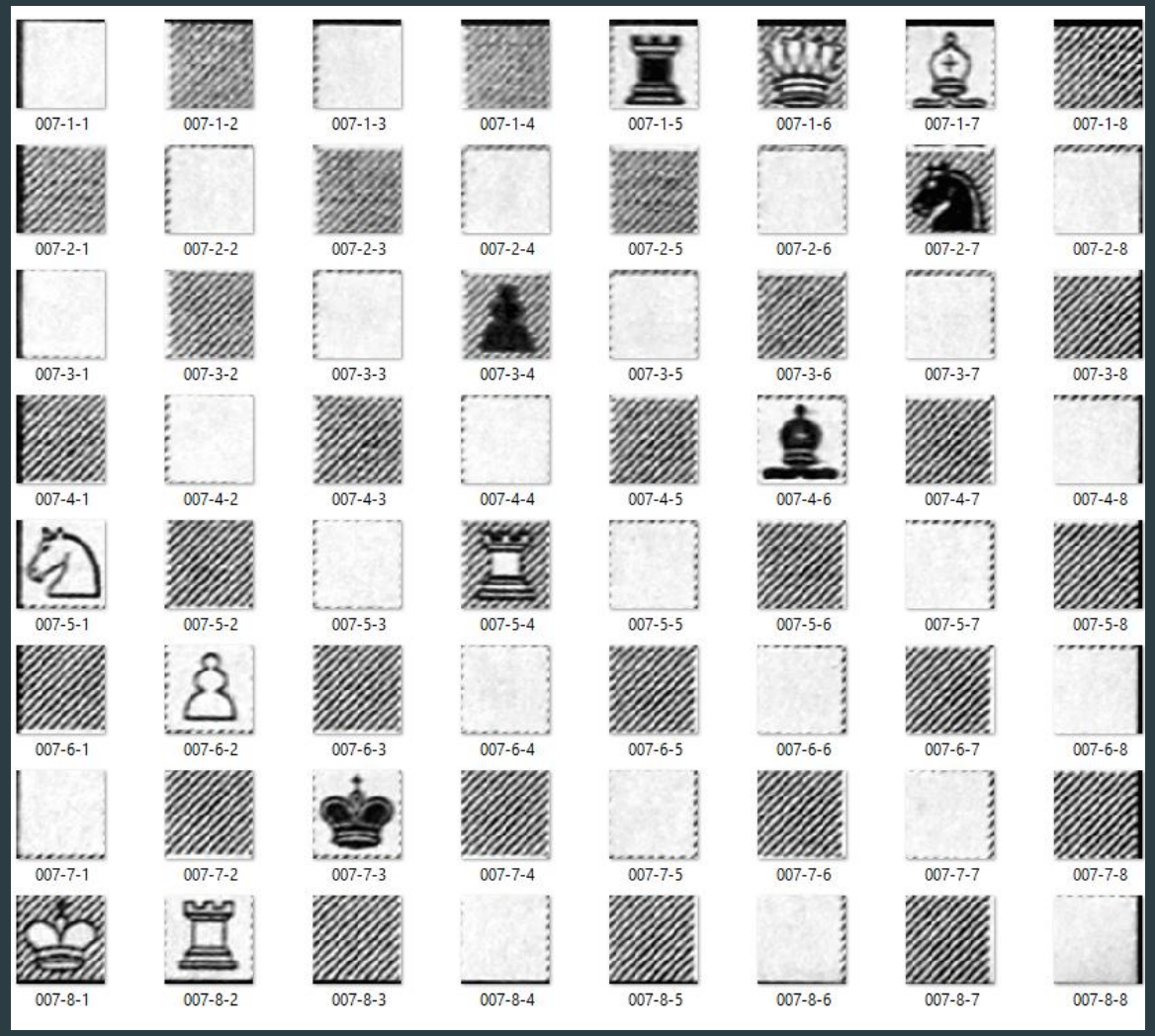
- The dataset is missing the black queen
- The number of training images for each label is not proportioned

- The trained model could be overfitted, and not suitable for general-purpose usage, expanding the dataset can help

- The dataset isn't all-around well-formatted, some features may be influenced



FEN prediction:



```
"FEN: 4rQBl/6nl/3p4/5b2/N2R4/lP6/2k5/KR6/"
```

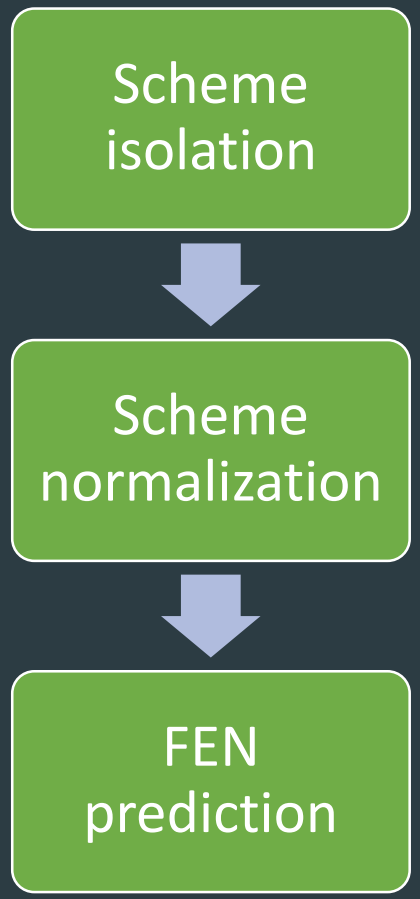
Considerations:

Assumptions:

- The schemes are always upright
- The schemes are reasonably squared

Results analysis:

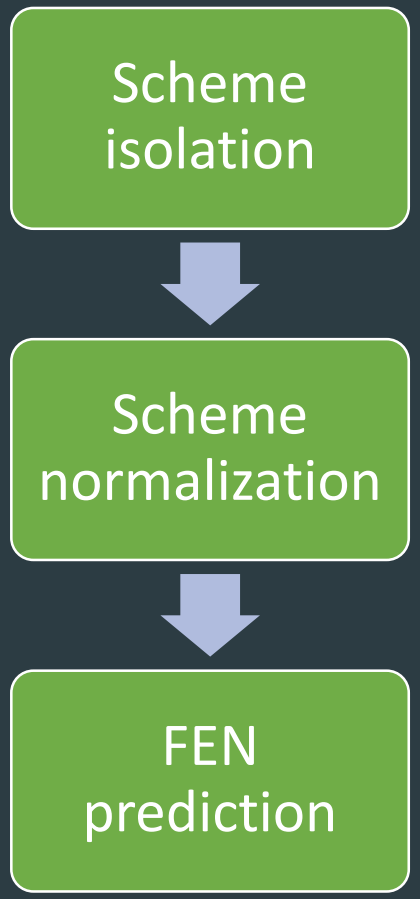
- Dataset: 48 photos
- Overall: 45/48 (93.75%) successfully processed photos



- 46/48 (96%) schemes extracted correctly (018 and 040 didn't pass)
- 45/46 (98%) schemes correctly normalized (039 didn't pass)
- 40/40 + 5/5 (100%) FEN correctly predicted

Results analysis – Execution time:

- Execution time for one photo: 18.3831s



- 1.9835 seconds
- 0.026893 seconds
- 16.372734 seconds (0.244997s * 64 + 0.6929s)

Results considerations - Conclusion:

- Time performance can be definitely improved:
  - Parallelize operation (parfor loop)
  - Reduce IO operations
  - Rationalize code
- Pipeline can be improved:
  - A data-driven approach for the scheme isolation can make it more robust to different photo conditions (Scheme classifier)
  - A scheme orientation classifier can make the pipeline more flexible
- The codebase can be improved:
  - Better design/guidelines
  - Rationalize code