Rubik’s cube state recognition

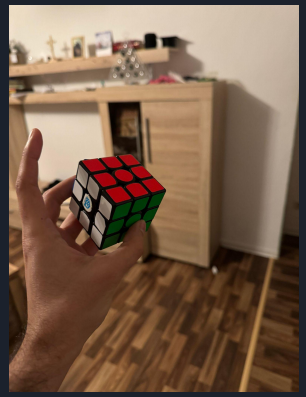
Liam Farhan - MIP

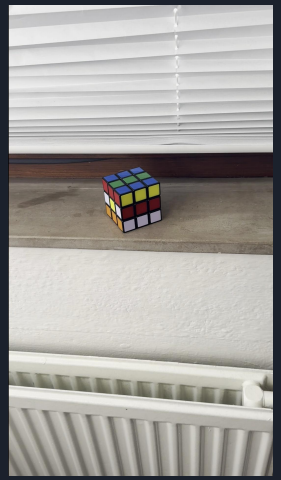
Approach

- Data Acquisition
- Train a CNN with model transfer
- Isolate faces
- Segment color stickers
- Obtain a representation of the cube state

Data acquisition

- Self taken images (ca. 40)
- Images downloaded from the internet (ca. 600)
- Video captured motion tracked images (ca. 400)





Data acquisition

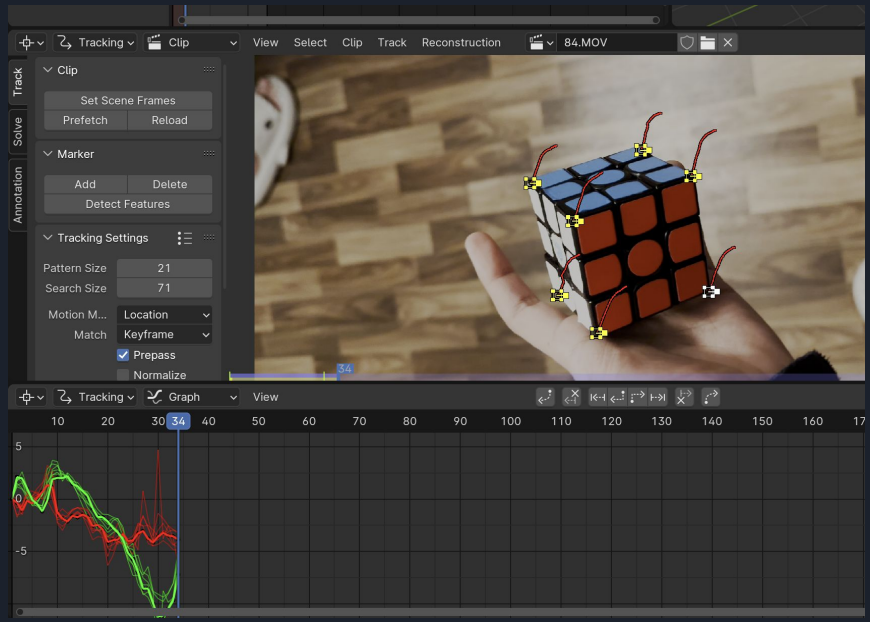
Labeling:

- Done per hand using Fiji
- Highlight all visible 7 corners

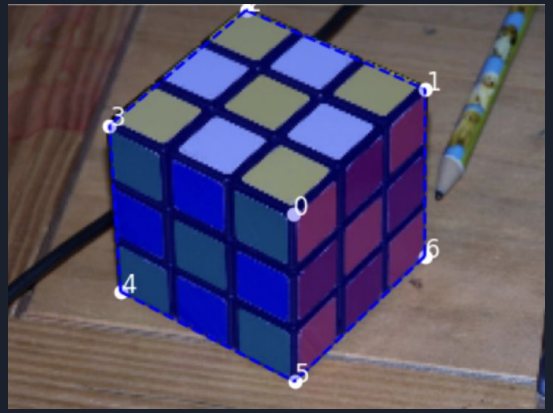
Data acquisition

Video captured motion tracked images:

- Take a video of a rubik's cube
- Move and rotate camera and cube
- Insure three faces are always visible
- Use blender to track the 7 corners of the cube
- Export tracking data using a script



Data preprocessing



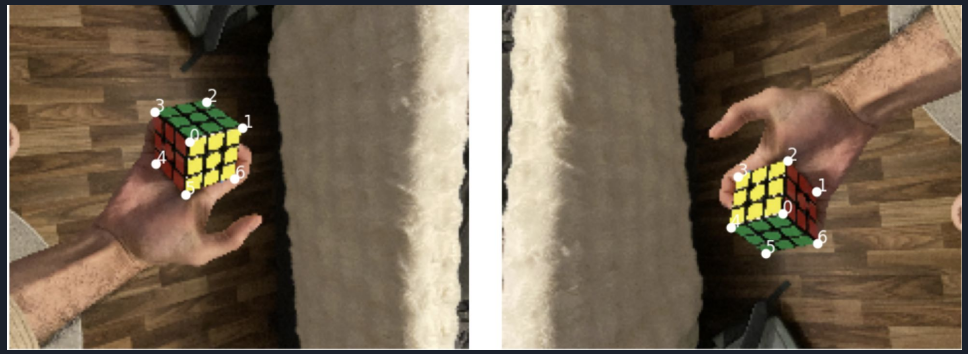
Sorting labels:

- Most important task in the project
- Isolate inner point by the fact it does not lay on the convex hull
- Sort the remaining points in a counter clockwise ascending order

Faces:

- Face 1 = (0,1,2,3)
- Face 2 = (0,3,4,5)
- Face 3 = (0,5,6,1)

Data preprocessing



Insure point quartets form faces:

- To avoid the issue in the picture
- Find the center of mass of a face (average coordinate of the face corners)
- If it does not lay near the intersection point of the diagonals
  Reorder the corners

Data preprocessing

Increase the amount of available data:

- Flip horizontally
- Flip vertically
- Flip diagonally
- Flip anti diagonally

Model Training

Data split:

- Video data is included only in training data
- Reflected versions are added only after training/test split
- Total number of training data points: 2207
- Total number of test data points: 750

Transfer learning:

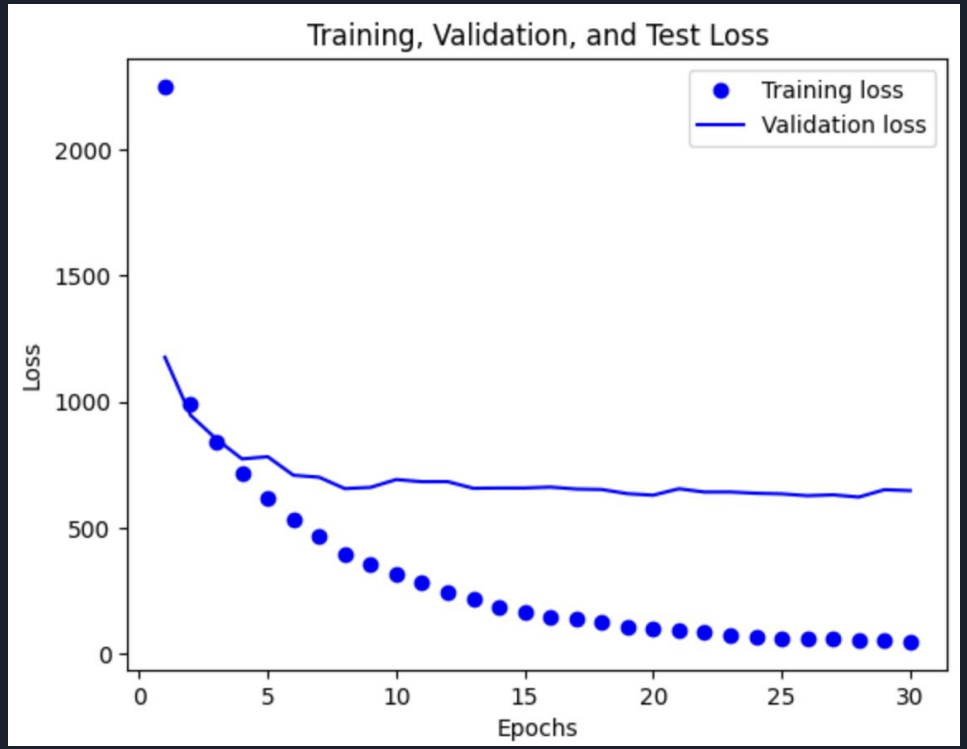
- Start with the VGG16 model trained on imagenet
- Add additional layers to extract the corner coordinates

Model Training

Model design: (outer layers)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 7, 7, 256)         1179904

 conv2d_1 (Conv2D)           (None, 7, 7, 128)         295040

 max_pooling2d (MaxPooling2  (None, 3, 3, 128)         0
 D)

 flatten (Flatten)           (None, 1152)              0

 dense (Dense)               (None, 64)                73792

 dense_1 (Dense)             (None, 14)                910

=================================================================
Total params: 1549646 (5.91 MB)
Trainable params: 1549646 (5.91 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

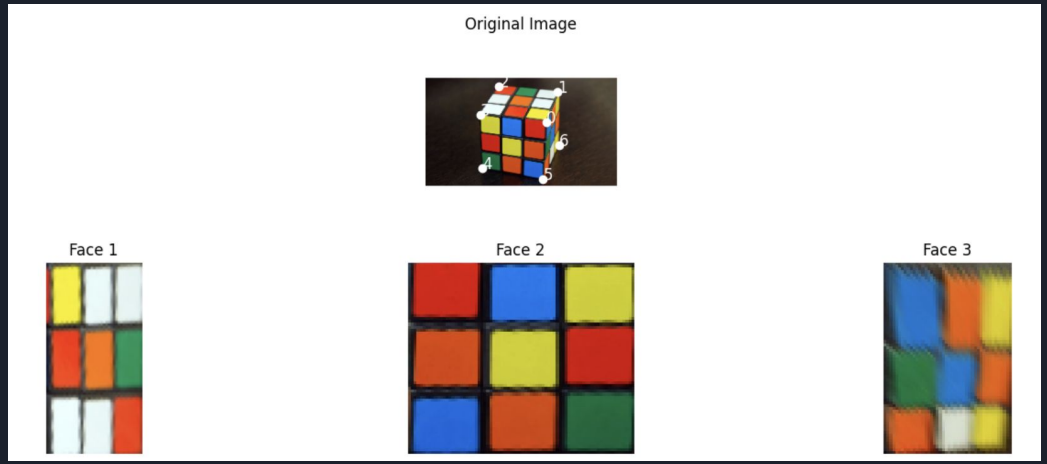
Model Training



Reconstructing the state

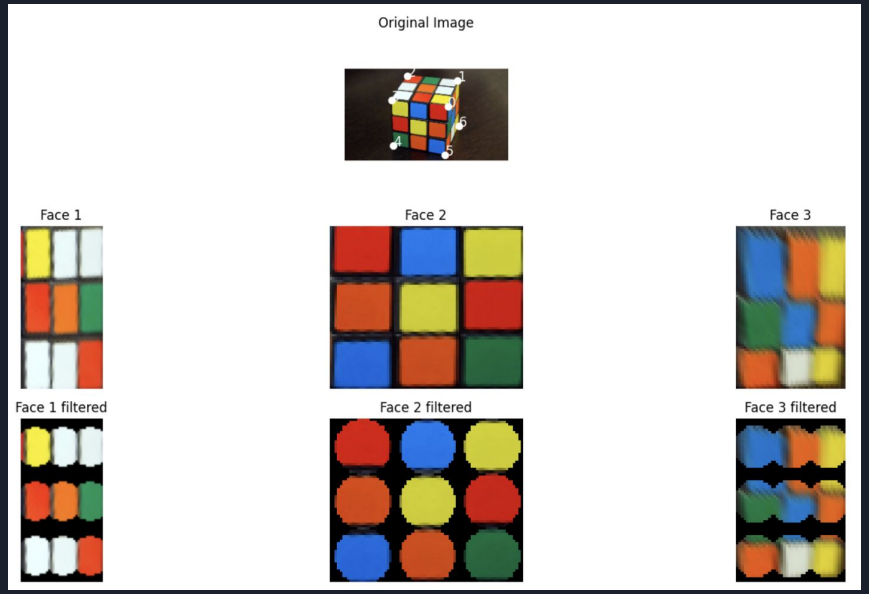
Use corner data to isolate the cube faces:

- Employ perspective projection



Reconstructing the state

Add receptors at fixed locations:



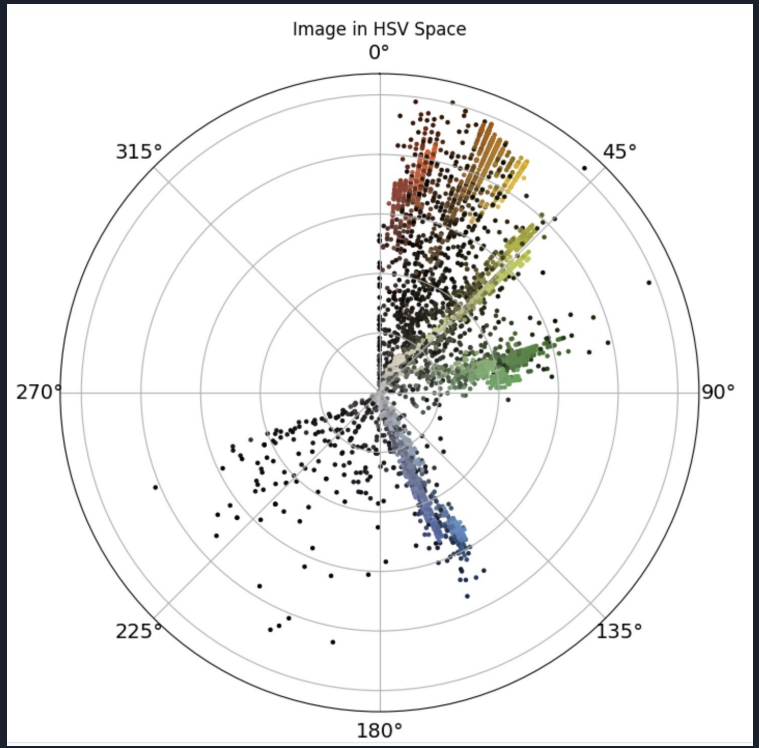
Reconstructing the state

Preform color segmentation

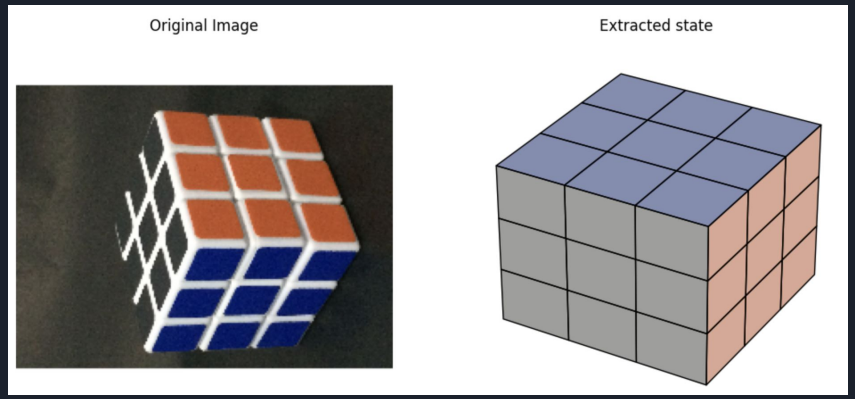
- In HSV space
- Use K Means to cluster the pixels in 7 color categories (sticker colors + background)
- Include all faces in the training

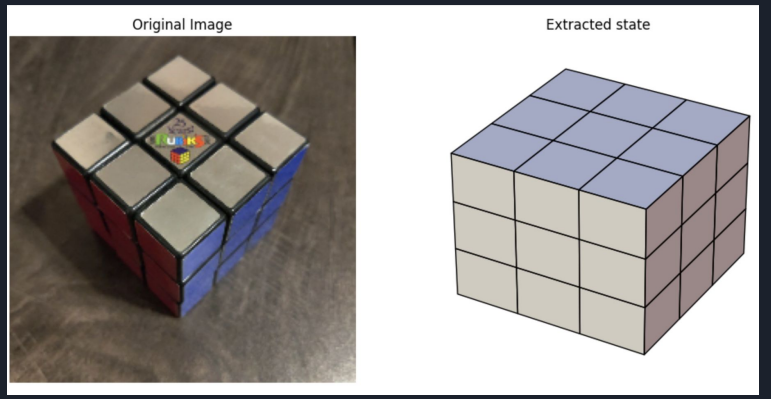
Reconstructing the state

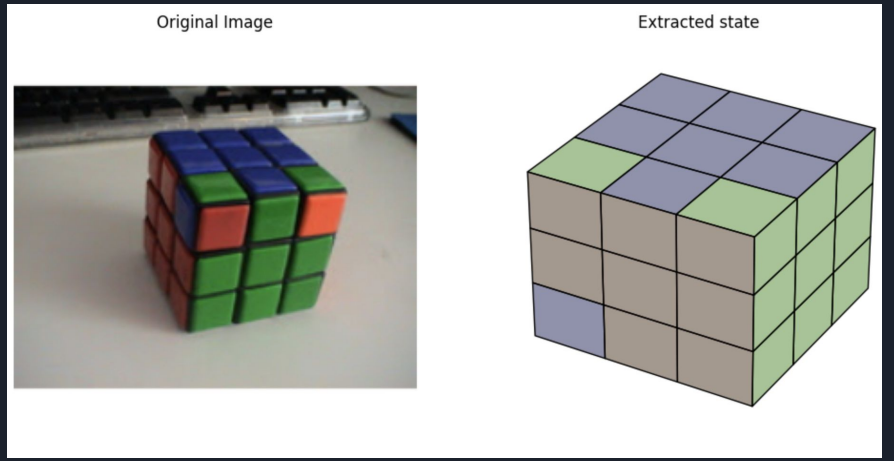
If a majority of the pixels in a receptive field belongs to a cluster classify the corresponding Sticker with said category.



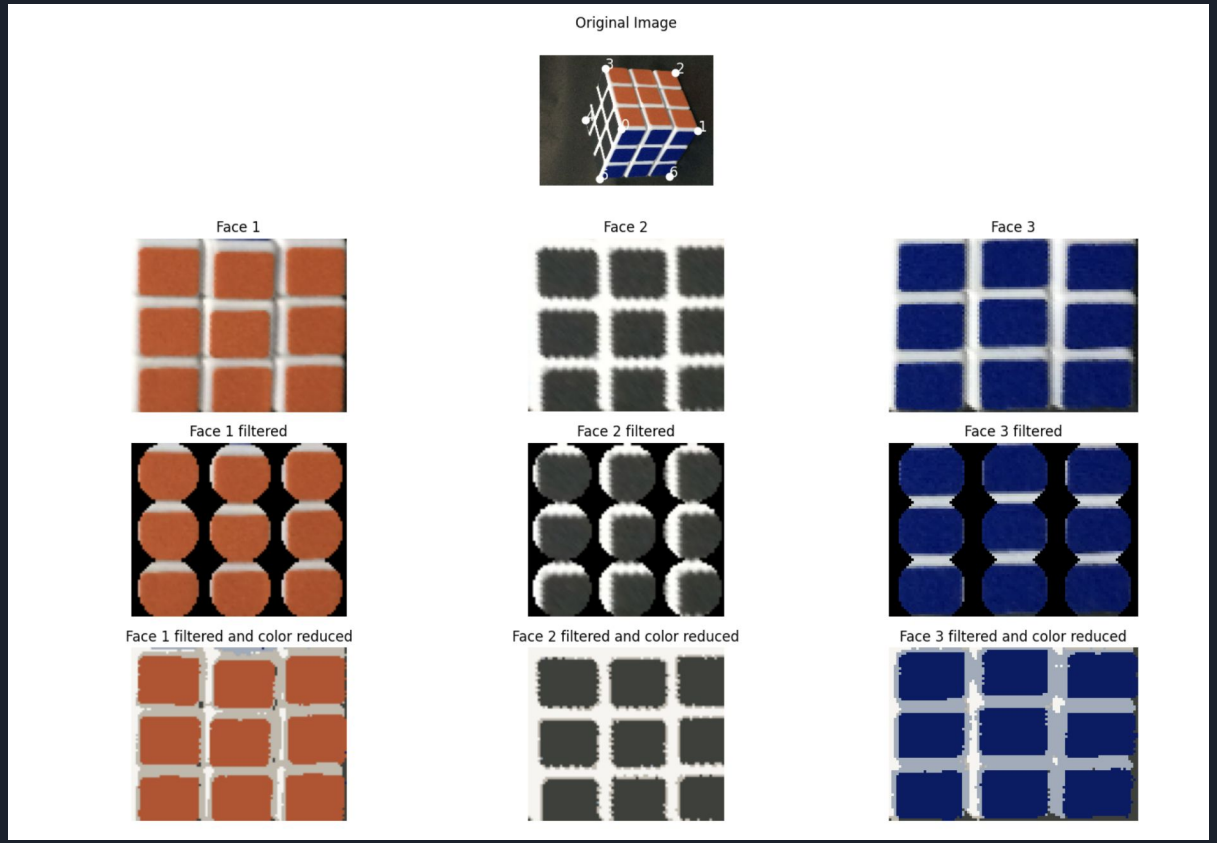
Some success



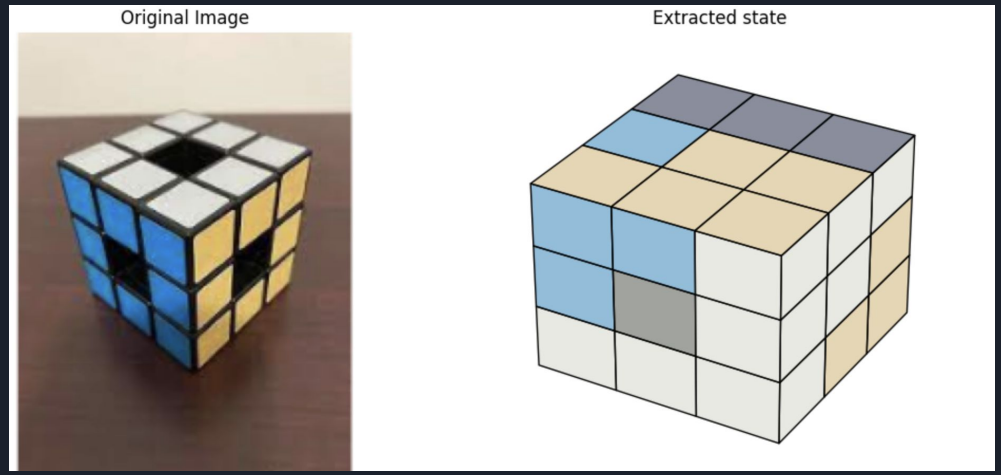




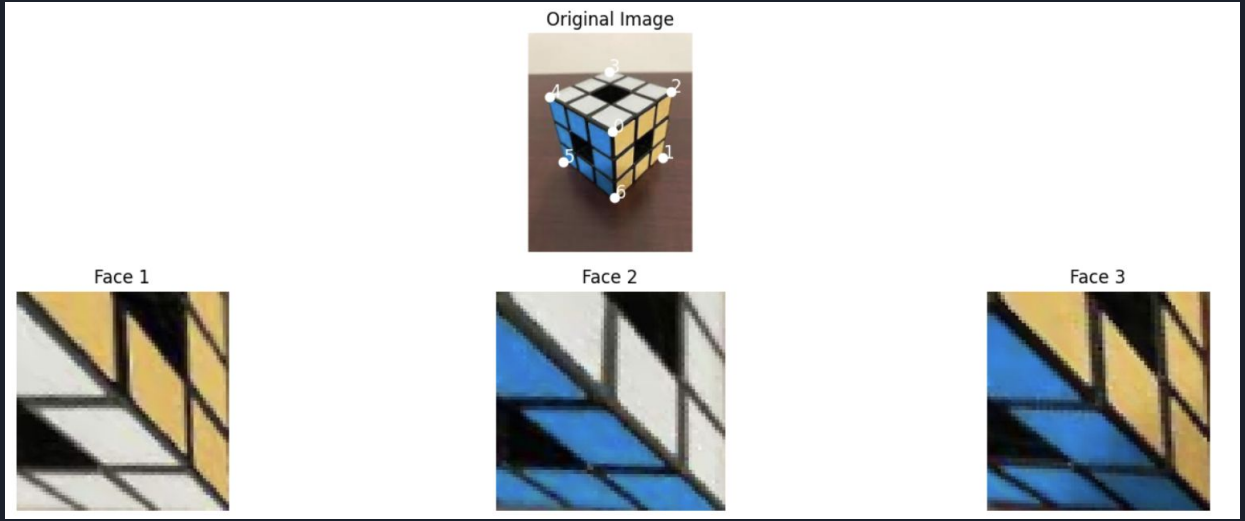
Some success



Some errors



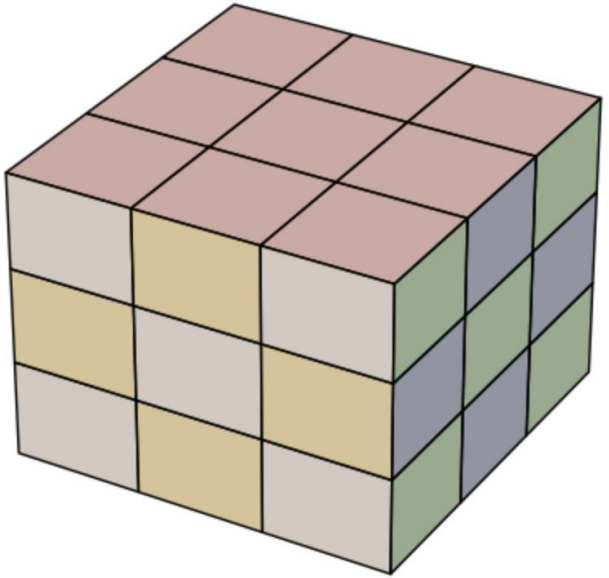
Some errors



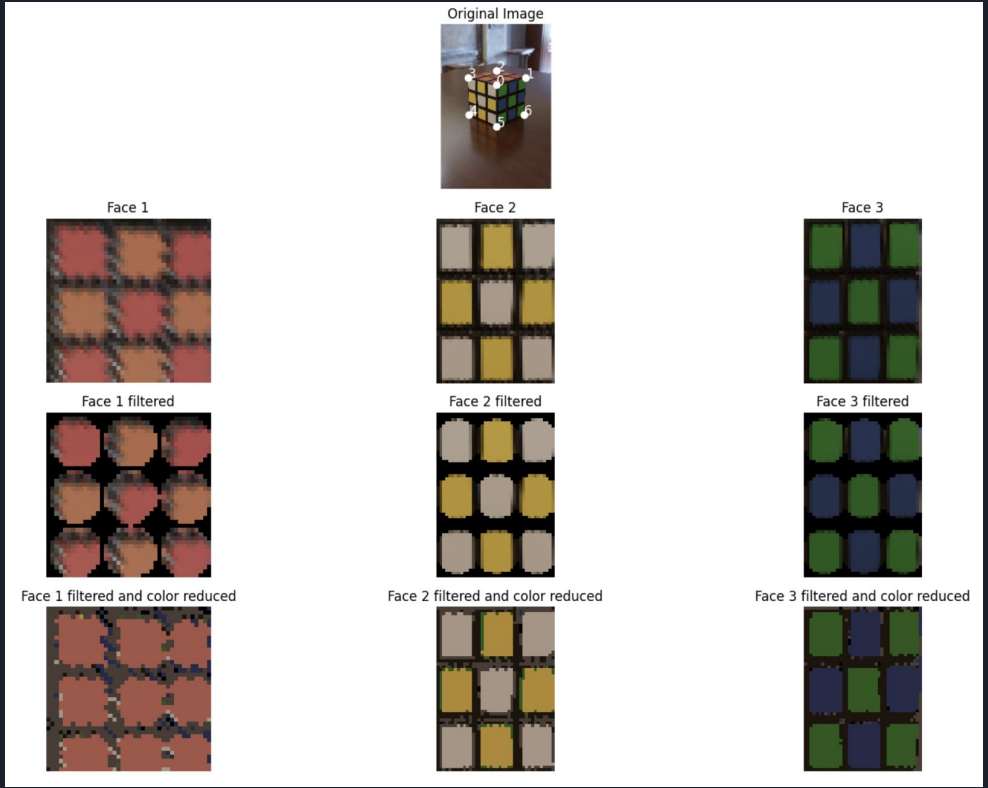
Some errors

Original Image

Extracted state



Some errors


Original Image
Face 1
Face 2
Face 3
Face 1 filtered
Face 2 filtered
Face 3 filtered
Face 1 filtered and color reduced
Face 2 filtered and color reduced
Face 3 filtered and color reduced

Questions?

Code accessible on Gitlab:

https://gitlab.gwdg.de/mohamad.alfarhan/mip_cube_state_recognition