Automated Physical Key Extraction from Images

Rory Smith, supervised by Dr Tilo Burghardt

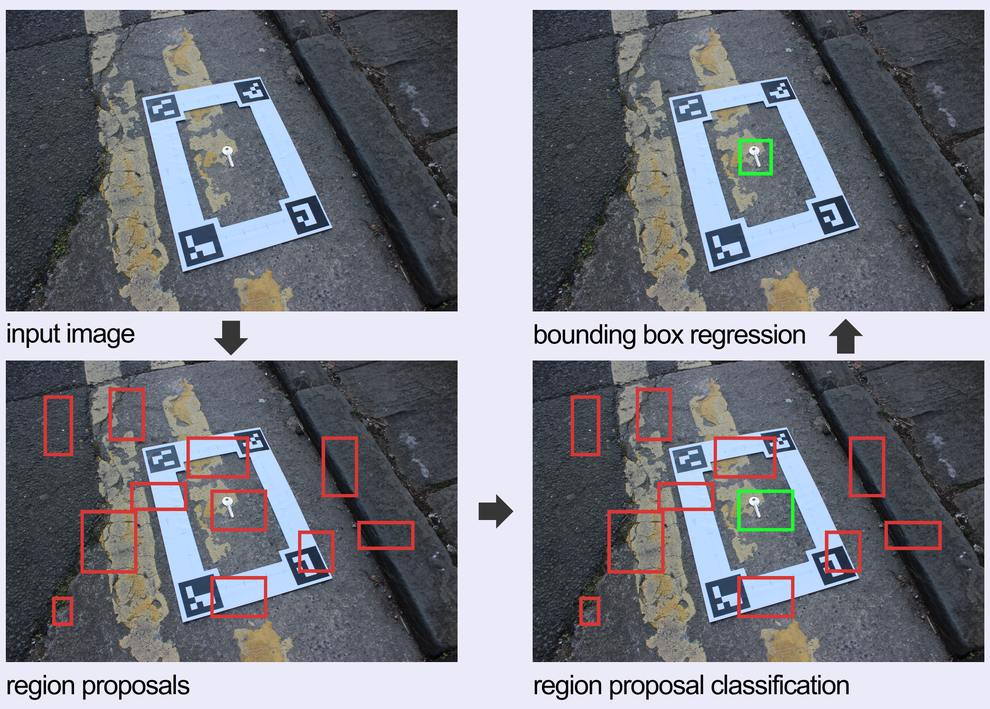
University of Bristol, Department of Computer Science

Motivation

People tend to guard their online passwords with care - why then, do we leave our car or **house keys** lying around in the public for anyone to see... and copy? This project aims to explore the possibility of copying keys from photos, in a **fully automated** way. With recent leaps in the effectiveness of **convolutional neural networks** (CNNs) and GPUs, the ability for computers to learn patterns from images has increased dramatically. We aim to harness these gains in CNNs to first detect, and then extract instances of keys. We hope in doing so, we will help demonstrate the need for higher security and conscientious protection of one's own keys.

1. Object Detection

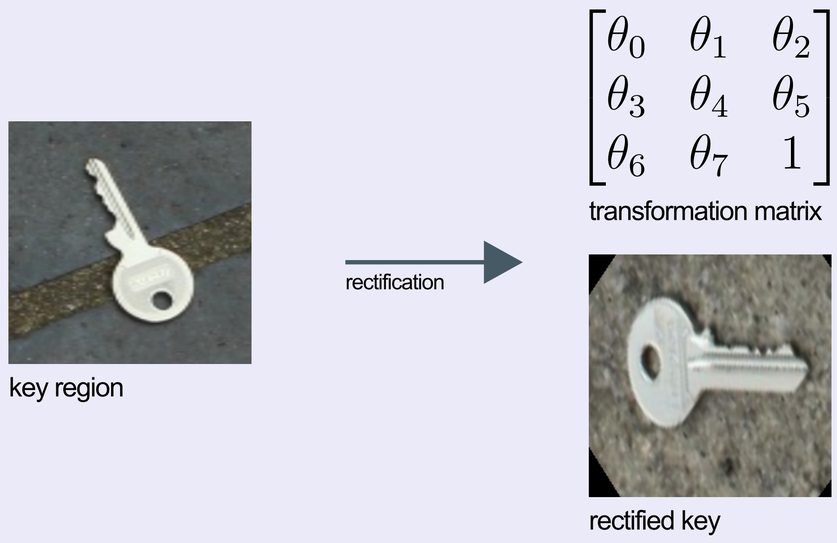
The first stage of the pipeline consists of an object detection network trained to locate keys in an image. For this, we base our architecture off **Faster - RCNN**. Images are input with ground truth bounding boxes containing keys. The architecture then produces region proposals and uses the feature maps output by **Resnet - 101** to decide whether a key is contained within a region. The region proposals are then regressed to more accurately bound the keys.



2. Key Rectification

After regions containing keys are detected, we use a **Spatial Transformer Network** (STN) to learn the transformation matrix resulting in a rectified view of the key. This learns 8 of the 9 parameters from a 3x3 transformation matrix in order to account for translation, rotation, scaling and perspective.

Our data collection method allows us to provide a transformation matrix ground truth for every image allowing us to perform **regression** with a set of very accurate data. We normalise the pose of the keys so that the masking ground truths need not be aligned to each key in the dataset - instead the rectified keys are aligned to a small set of mask ground truths, preventing the need for annotating every image with a ground truth mask.



Data

Much of deep learning's success comes from ingesting **large amounts of data**. This data needs to be appropriately labeled so that the network can learn what the appopriate response is, given some input image.

To generate data for all stages of the pipeline in a **semi-unsupervised** way, we created a 4-marker frame which we place around keys before taking pictures. Our software can find this frame in each photo, locate the key and find the transformation matrix that rectifies the key. Not only does this mean we do not need to manually annotate bounding boxes, but it gives us a much higher level of accuracy for our transformation matrix than we could achieve by hand.

We also experimented with using **synthetically generated** datasets to train key rectification. Based off a set of top-down viewing angle photos of our key set, we then applied geometric and visual effects, and composited a random background layer. Although this allowed us to generate a much larger dataset ~20,000 samples, this did not translate well to the real world dataset. We believe this to be a consequence of the subtle lighting effects on keys present in the real dataset, which is unobtainable in the synethic set, unless calculated and rendered using 3D models.

marker frame system for data labeling

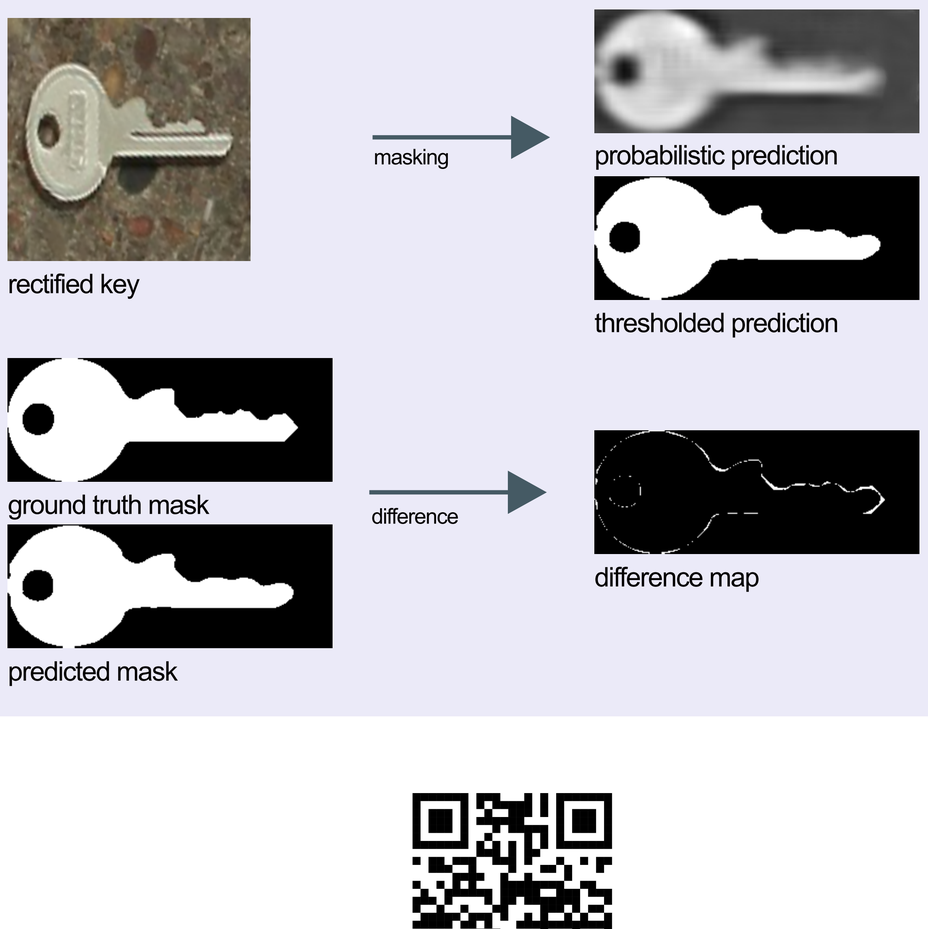
synthetically generated data

3. Key Masking

The final stage consists of applying the masking head from **Mask - RCNN** to the rectified key samples. By passing the image rectifications through a second STN, we can allow the network to align each key rectification to the key's binary mask in an **unsupervised** manner.

The masking head applies several convolutional layers and a single deconvolutional layer making it a **fully convolutional network** (FCN). It outputs a somewhat blurry mask, but our **average binary cross-entropy loss** means that individual pixels can be thresholded to their nearest class with a sufficiently high level of accuracy (as seen below in the difference map).

This mask can then be used to generate a printable 3D model of the key when combined with the key's known keyway (side view). Future work may include learning of the keyway from images to produce a complete solution.



University of BRISTOL