IDENTIFICATION OF DEFECTIVE PRODUCTS WITH A CONVOLUTIONAL AUTOENCODER

HIBÁS TERMÉKEK FELISMERÉSE KONVOLÚCIÓS AUTOENKÓDERREL

CH1KEN_RUN

Galacz Barnabás(D133RO)

Simon Zalán (IOL89K)

Szabó Zoltán (BWLQ1Y)

Table of Contents

1 Abstract

1.1 English

With the utmost importance of successful detection of defective products in mass production, the need for more precise detection technologies have arisen. One of these technologies are deep neural networks, more precisely convolutional autoencoder architectures. These architectures have the advantage, that their requirement for training data on anomalous object are nonexistent, therefore bypassing the limitations of regular classification networks. In our work, we took on the challenge to develop, optimize and evaluate a convolutional autoencoder network for the successful identification of defective products. We built on the work of Bergmann et al. and used the MVTec dataset.

1.2 Hungarian

A selejtes termékek kiszűrésének problémájának rendkívüli fontossága miatt megnövekedett az igény a precíziós észlelési technológiák iránt. Az egyik ilyen technológia a mély tanulási, ezen belül is a konvolúciós autoencoder architektúrák. Ezen architektúrák előnye, hogy nincs szükség a hibás termékek, és azon belül is a különböző hibák egyenkénti betanítására, ezáltal lényegesen hatékonyabbak tudnak lenni a hagyományos, klasszifikációra használt hálóknál. A projekt során vállaltuk a kihívást, és egy már elkészített autoencodert fejlesztettünk tovább, és optimalizáltunk az általunk vizsgálni kívánt termékre. Bergmann és társainak a munkáját felhasználva az MVTec adatbázisában összegyűjtött képanyaggal dolgoztunk.

2 Introduction

At our current age almost all of our goods are being mass produced. Mass production however needs to have strict quality assurance. The vast majority of current state-of-the-art production technologies are so advanced that defective products rarely appear on the production lines. Because of this, regular quality assurance methods have a hard time identifying and withholding these products. Therefore, factories are interested in more advanced identification technologies, many of which are provided by the IT field. One of the most sought after technology is the usage of computer aided defective product identification, which mostly are built on some type of artificial intelligence, for example neural networks. The problem with regular classification neural networks arises from the fact, that defective products rarely appear, and these defects can take on several different forms. Therefore, training a regular classification network can be challenging. As a solution to this problem Convolutional Autoencoders (CAE) are being developed. This architecture allows us to identify anomalies on the given image of the product with minimal training data on the anomaly itself. We took on the challenge to develop a CAE architecture, optimize it, then successfully identify the vast majority of anomalous object. In our work we built on the work of Bergmann et al. and used the MVTec Anomaly Detection Dataset [1].

3 Literature Review

"For the evaluation of methods that segment anomalies in images, only very few datasets are currently available to the public. Many of them are limited to the inspection of textured surfaces or focus on novelty detection in multi-class semantic segmentation scenarios".

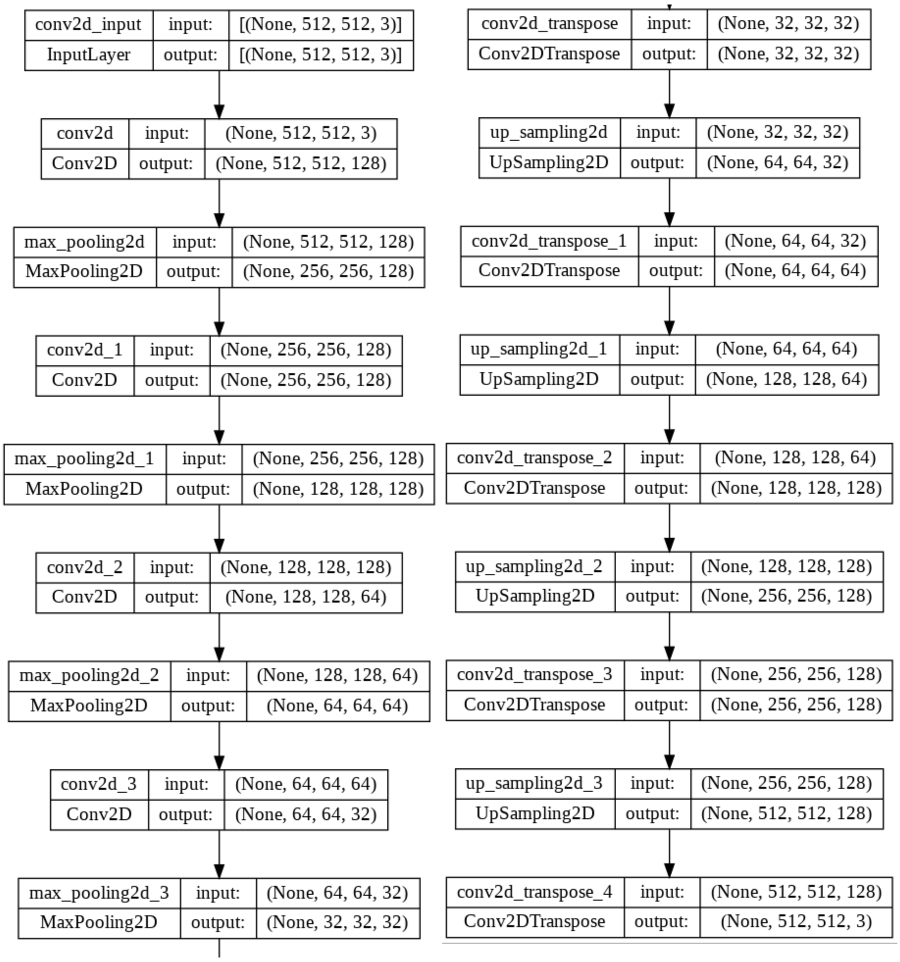
There is a wide range of methods and approaches that have been suggested to solve the challenge of unsupervised anomaly detection, however these solutions are quite diverse.

There have been a lot of different databases available, but they all lacked something. A dataset was specifically designed by Wieler and Hahn for optical inspection of textured surfaces, but the pictures were artificially generated, that can only be taken as an approximation to the real world, and the dataset was poorly labeled and the variance between the textures and generated was data is insignificant. [2-5]

4 Neural network Architecture, Testing and Performance Metrics

4.1 System Architecture

Our neural network follows the convolutional autoencoder model which means the network first encodes the image in a layer with smaller dimensions. After that it tries to replicate the image and as the network is only trained on images with no anomalies it has a hard time reproducing them. This way if we send an image with an anomaly through the network and compare it with the original the anomaly should mostly disappear, and it will not resemble the original image. This way we can filter out anomalies. The figure below shows the complete architecture of the network.



4.2 Implementation

. For the network we added four convolutional layers and a max pooling on one side, and it went from 128 filters down to 32 and after we decoded it back to 128 with a mirrored kind of neural network. After that we used the adam optimizer. We did not have many images, so we thought the network for 100 epochs and set 16 s our batch size. To match these numbers, we set the steps per epoch to 250/16 and the validation steps to 2.

After that we extract the encoder network with trained weights to get the compressed output (latent space) of the input image. The compressed output is then used to calculate the KDE. Then we Calculate the KDE using sklearn. Get the encoded output of input images, Flatten the encoder output because KDE from sklearn takes 1D vectors as input and fit the KDE to the image latent data. After that we created a function that checks if the images contain anomalies. We do this by checking the images density and reconstruction error, which are then compared to the ones that do not have anomalies.

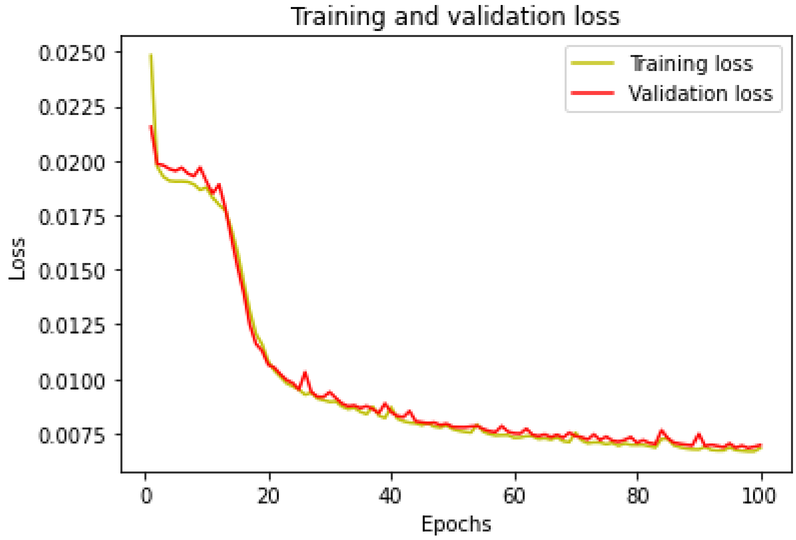
4.3 Preparing Dataset

The dataset was provided by MVtec, as it is publicly available on their website. Different types of products are contained in the dataset such as nails, walnut, medicine pills, cables etc. We have chosen to work with pictures of carpets, as their surface has a homogenous pattern, and the pictures contain no background which is unnecessary data for our use case.

We then sorted the pictures into different working directories after confirming that each of them has the number of rows and columns of pixels.

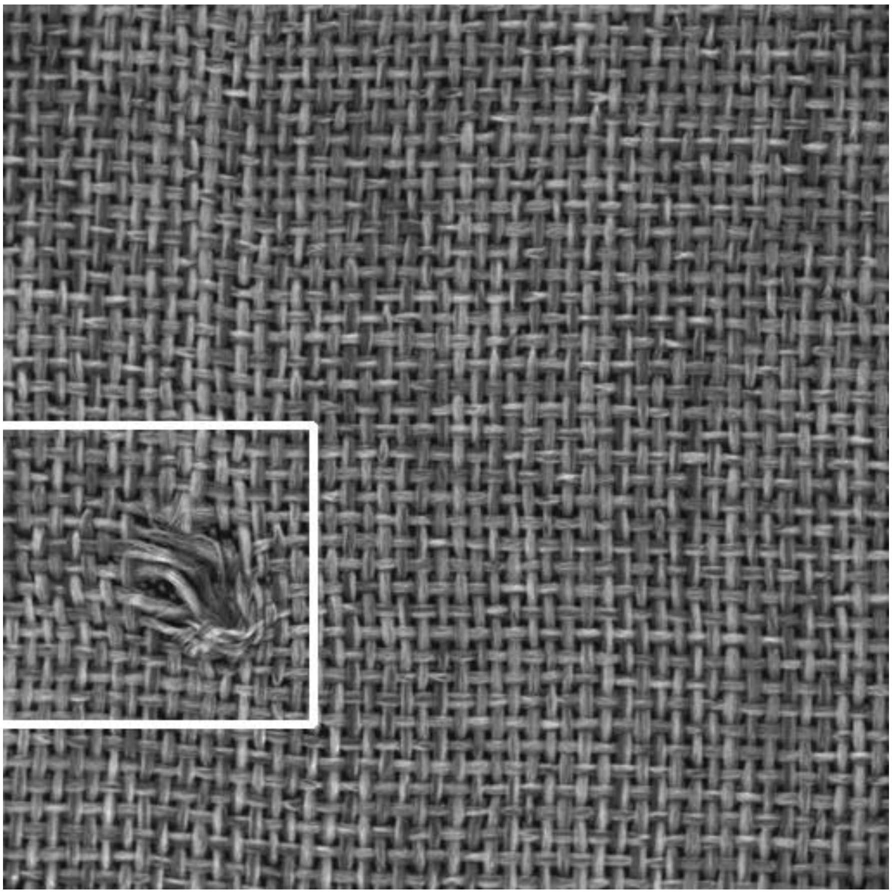
4.4 Training

The model is trained for 300 epochs, using the standard set per epoch number which is (Total Number of Training Samples / Batch Size). The same went for the validation steps just with the number of validation samples. Due to the small number of samples we used 16 as a batch size. We came to this number by trial and error as we tried larger number that worked less efficiently. The number of epochs could be larger but due to a larger bottle neck it was too good at reconstructing anomalies. With a smaller bottleneck it and larger epoch number it was better at catching anomalies but couldn’t locate the location of the anomaly using the image subtracting method.



4.5 Evaluation

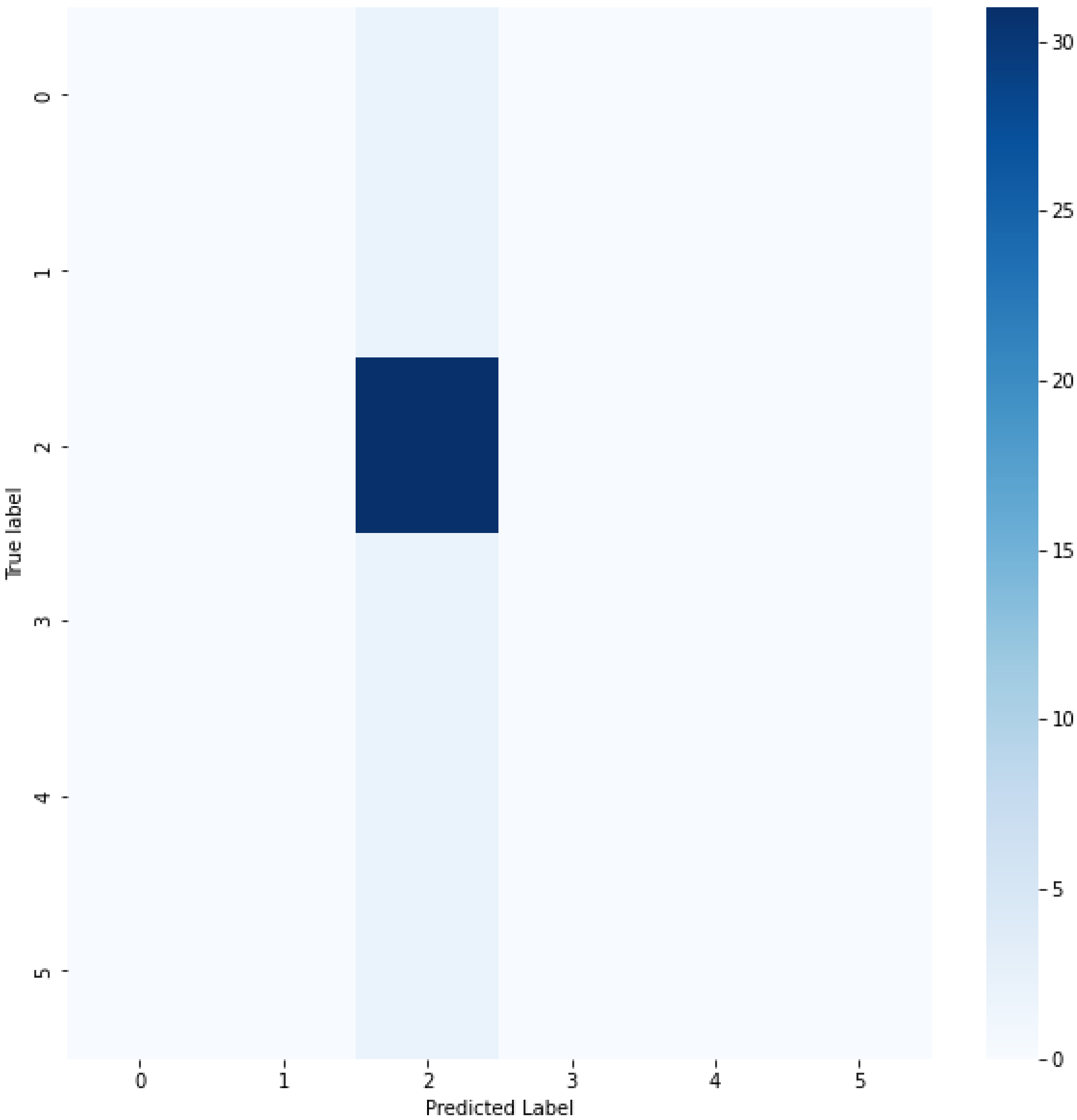
During the evaluation phase of the work we measured how well our network performs and compared it with a regular convolutional neural network. The autoencoder network can successfully identify anomalous images, as well as highlight the part that is defective. For an example look at the image below:



The network does the anomaly recognition by trying to reconstruct the given image. If the reconstruction proceeds with an error value below the given threshold, then the network classifies the image as an anomaly free one, if the error exceeds the threshold, then the given image must be a defective one.

The rectangle over the defective part is placed by the reconstruction information. By giving the network a defective image, it cannot reproduce it, this way we will have two separate images with the anomaly itself being the main difference. Then we compare the two images, and a rectangle is placed on the part with the highest amount of differences.

Our work shows the reason why we prefer the use of autoencoder networks in the process of identifying defective objects over regular convolutional neural networks. The latter type of architecture requires the use of extensive amount of training data over all types of anomalies to be identified. However, this amount of training data is usually unattainable. On the image below you can see that the CNN architecture can't really identify any anomalies, it labels all the images given to it as 'good' even the anomalous ones (for label numbering check the given code).



4.6 Testing

We conducted testing throughout the development of the network to confirm every newly written function works as it is supposed to.

After the whole network was thought to be finished, then we used pictures - that have been taken of defective products - to test how accurate our network is.

Around 20 different images that are known to have anomalies on them were fed into the software. Nearly all the anomalies were recognized, however, their exact location was not accurately always shown. This is probably the most important part of the code, that needs to be improved in the future.

5 Plans for the Future

In the future we could improve on the model with adding more images to database even with just rotating the ones available. That would allow us to run more epochs and have better reconstructive abilities.

The network could also be trained to recognize other products too.

References

1. Bergmann, P., et al., *The MVTec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection.* International Journal of Computer Vision, 2021. **129**(4): p. 1038-1059.
2. Lien, C.-C. and Y.-D. Chiu, *A Defect-Inspection System Constructed by Applying Autoencoder with Clustered Latent Vectors and Multi-Thresholding Classification.* Applied Sciences, 2022. **12**(4): p. 1883.
3. Jung, J., et al., *A Comprehensive Real-World Photometric Stereo Dataset for Unsupervised Anomaly Detection.* IEEE Access, 2022. **10**: p. 108914-108923.
4. Artola, A., et al. *Unsupervised Variability Normalization For Anomaly Detection*. in *2021 IEEE International Conference on Image Processing (ICIP)*. 2021. IEEE.
5. Böttger, T. and M. Ulrich, *Real-time texture error detection on textured surfaces with compressed sensing.* Pattern Recognition and Image Analysis, 2016. **26**(1): p. 88-94.