Few shot learning for anomaly detection
Bachelor Thesis for AI

Lukas Heiligenbrunner

Abstract — Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo.

Lukas Heiligenbrunner
Johannes Kepler University, Linz, Austria, email: lukas.heiligenbrunner@gmail.com

Contents

1 Introduction

1.1 Motivation

Anomaly detection has especially in the industrial and automotive field essential importance. Lots of assembly lines need visual inspection to find errors often with the help of camera systems. Machine learning helped the field to advance a lot in the past. PatchCore and EfficientAD are state of the art algorithms trained only on good data and then detect anomalies within unseen (but similar) data. One of their problems is the need of lots of training data and time to train. Few-Shot learning might be a suitable alternative with essentially lowered train time.

In this thesis the performance of 3 Few-Shot learning algorithms will be compared in the field of anomaly detection. Moreover, few-shot learning might be able not only to detect anomalies but also to detect the anomaly class.

1.2 Research Questions

1.2.1 Is Few-Shot learning a suitable fit for anomaly detection?

Should Few-Shot learning be used for anomaly detection tasks? How does it compare to well established algorithms such as Patchcore or EfficientAD?

1.2.2 How does disbalancing the Shot number affect performance?

Does giving the Few-Shot learner more good than bad samples improve the model performance?

1.2.3 How does the 3 (ResNet, CAML, pmf) methods perform in only detecting the anomaly class?

How much does the performance improve if only detecting an anomaly or not? How does it compare to PatchCore and EfficientAD?

1.2.4 Extra: How does Euclidean distance compare to Cosine-similarity when using ResNet as a feature-extractor?

I've tried different distance measures → but results are pretty much the same.

1.3 Outline

todo

2 Material and Methods

2.1 Material

2.1.1 MVTec AD

MVTec AD is a dataset for benchmarking anomaly detection methods with a focus on industrial inspection. It contains over 5000 high-resolution images divided into fifteen different object and texture categories. Each category comprises a set of defect-free training images and a test set of images with various kinds of defects as well as images without defects.

2.2 Methods

2.2.1 Few-Shot Learning

Few-Shot learning is a subfield of machine-learning which aims to train a classification-model with just a few or no samples at all. In contrast to traditional supervised learning where a huge amount of labeled data is required is to generalize well to unseen data. So the model is prone to overfitting to the few training samples.

Typically a few-shot leaning task consists of a support and query set. Where the support-set contains the training data and the query set the evaluation data for real world evaluation. A common way to format a few-shot leaning problem is using n-way k-shot notation. For Example 3 target classeas and 5 samples per class for training might be a 3-way 5-shot few-shot classification problem.

A classical example of how such a model might work is a prototypical network. These models learn a representation of each class and classify new examples based on proximity to these representations in an embedding space.

The first and easiest method of this bachelor thesis uses a simple ResNet to calucalte those embeddings and is basically a simple prototypical netowrk. See

2.2.2 Generalisation from few samples}

2.2.3 Patchcore}

%todo also show values how they perform on MVTec AD

2.2.4 EfficientAD

todo stuff (Roth, Pemula, Zepeda, Schölkopf, Brox, Gehler 2022) todo stuff (Batzner, Heckler, König 2024)

2.2.5 Jupyter Notebook

A Jupyter notebook is a shareable document which combines code and its output, text and visualizations. The notebook along with the editor provides a environment for fast prototyping and data analysis. It is widely used in the data science, mathematics and machine learning community.

In the context of this practical work it can be used to test and evaluate the active learning loop before implementing it in a Dagster pipeline. (2024a)

2.2.6 CNN

Convolutional neural networks are especially good model architectures for processing images, speech and audio signals. A CNN typically consists of Convolutional layers, pooling layers and fully connected layers. Convolutional layers are a set of learnable kernels (filters). Each filter performs a convolution operation by sliding a window over every pixel of the image. On each pixel a dot product creates a feature map. Convolutional layers capture features like edges, textures or shapes. Pooling layers sample down the feature maps created by the convolutional layers. This helps reducing the computational complexity of the overall network and help with overfitting. Common pooling layers include average- and max pooling. Finally, after some convolution layers the feature map is flattened and passed to a network of fully connected layers to perform a classification or regression task. Fig. 1 shows a typical binary classification task. (O'Shea, Nash 2015)

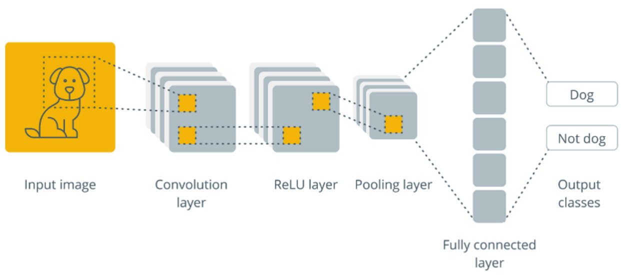

Fig. 1: Architecture convolutional neural network. (2024b)

2.2.7 RESNet

Residual neural networks are a special type of neural network architecture. They are especially good for deep learning and have been used in many state-of-the-art computer vision tasks. The main idea behind ResNet is the skip connection. The skip connection is a direct connection from one layer to another layer which is not the next layer. This helps to avoid the vanishing gradient problem and helps with the training of very deep networks. ResNet has proven to be very successful in many computer vision tasks and is used in this practical work for the classification task. There are several different ResNet architectures, the most common are ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152. (He, Zhang, Ren, Sun 2015)

Since the dataset is relatively small and the two class classification task is relatively easy (for such a large model) the ResNet-18 architecture is used in this practical work.

2.2.8 CAML

Todo

2.2.9 P>M>F

Todo

2.2.10 Softmax

The Softmax function Equation 1 (Liang, Wang, Lei, Liao, Li 2017) converts n numbers of a vector into a probability distribution. Its a generalization of the Sigmoid function and often used as an Activation Layer in neural networks.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{k} e^{z_k}} \text{ for } j = (1, ..., k) \tag{1}$$

The softmax function has high similarities with the Boltzmann distribution and was first introduced in the 19th century (Boltzmann 2012).

2.2.11 Cross Entropy Loss

Cross Entropy Loss is a well established loss function in machine learning. Equation eqref{eq:crelformal}cite{crossentropy} shows the formal general definition of the Cross Entropy Loss. And equation eqref{eq:crelbinary} is the special case of the general Cross Entropy Loss for binary classification tasks.

$$
\begin{aligned}
H(p,q) &= -\sum_{x\in\mathcal{X}} p(x)\log q(x)\\
H(p,q) &= -p\log(q) + (1-p)\log(1-q)\\
\mathcal{L}(p,q) &= -\frac{1}{N}\sum_{i=1}^{\mathcal{B}}(p_i\log(q_i) + (1-p_i)\log(1-q_i))
\end{aligned}
\tag{2}
$$

Equation $\mathcal{L}(p,q)$ eqref{eq:crelbinarybatch}cite{handsonaiI} is the Binary Cross Entropy Loss for a batch of size $\mathcal{B}$ and used for model training in this Practical Work.

2.2.12 Mathematical modeling of problem

3 Section Heading

(Batzner, Heckler, König 2024)

3.1 Subsection Heading

3.1.1 Subsubsection Heading

Paragraph Heading

Subparagraph Heading

Bibliography

Batzner, K., Heckler, L., & König, R. (2024,). *EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies.* https://arxiv.org/abs/2303.14535

Boltzmann, L. (2012). Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten. In F. Hasenöhrl (Ed.), *Wissenschaftliche Abhandlungen* (pp. 49–96). Cambridge University Press.

He, K., Zhang, X., Ren, S., & Sun, J. (2015,). *Deep Residual Learning for Image Recognition.*

Liang, X., Wang, X., Lei, Z., Liao, S., & Li, S. Z. (2017). Soft-margin softmax for deep classification. *International Conference on Neural Information Processing*, 413–421.

O'Shea, K., & Nash, R. (2015,). *An Introduction to Convolutional Neural Networks.*

Project Jupyter Documentation. (2024a,).

Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., & Gehler, P. (2022,). *Towards Total Recall in Industrial Anomaly Detection.* https://arxiv.org/abs/2106.08265

What are convolutional neural networks?. (2024b,).