Anomaly detection for automated quality control

Philip Montsho

22/08/2018

Aim:

Smart factories prioritize time-saving and automation, thus any reduction in the duration of quality control procedures translates directly into increased production.

This project investigates the benefits of using supervised and unsupervised machine learning algorithms to detect defects and anomalies that were caused during the production of nails. The classification performance of the different models will be analysed and a simple supervised model will be deployed via a REST API.

Using an entirely unsupervised algorithm that doesn't rely on labels to predict if a part is defective or not can reduce the deployment time, since the machine learning algorithm will be able to segment the images into different clusters automatically based on features it observed. These clusters would ideally consist of one main cluster which are the non-defective parts, then a few smaller clusters that each have the same defect or the same combination of defects and finally there would be a few random points that don't form a part of any of these clusters, which are anomalies.

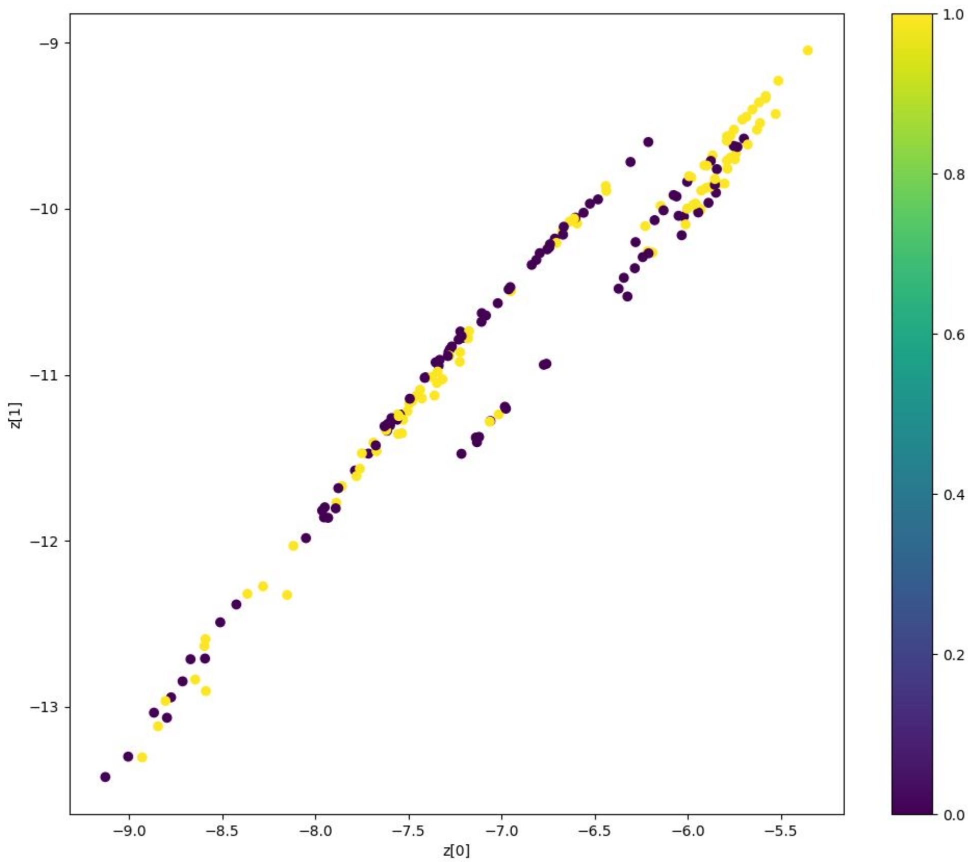
Variational Autoencoder Performance:

During training of the VAE the vae-loss used to describe the VAEs ability to reconstruct the original input from the 2D latent encoding, was incredibly high and did not decrease a significant amount as training continued. And after training the model for 10 epochs and a batch size of 10 the 2D latent space encoding didn't cluster the nails in a meaningful way that would make it possible to distinguish between the defective and non-defective nails. The results of the automated clustering approach using dbscan are no better than guessing.

2D latent space representation:

1 = non-defective

0 = defective



Classification report:

	precision	recall	f1-score	support
0	0.54	0.32	0.41	99
1	0.61	0.52	0.56	99
avg / total	0.58	0.42	0.48	198

Although the automated clustering approach had a very low performance, upon manual visual inspection the 2D latent space representation does show a separation between the defective and non-defective points. This separation could be the representation of the deformation of the nail since it is very suttle and the overall dispersion of the points across the 2D space could primarily be a representation of the translation and rotation of the nails.

An unsupervised learning model would work best in this case if it is entirely translation, rotation and scale invariant, which VAEs are not but some CNN VAEs are. And even at their best a supervised learning algorithm would still work better for a classification task.

Gradient Boosting Classifier Performance:

The following two sections will analyse the performance that was achieved using a supervised gradient boosting classifier. The sections will also compare the performance of the classifier that is trained on smaller images vs larger images. One of the most straight-forward concerns one may have when using/choosing a machine learning toolkit is the latency at which predictions can be made in a production environment. Although we would expect the model that was trained on higher quality images to be more accurate, we would also expect this model to take longer to make a prediction. The latency is not investigated here in more detail since it is not the focus of the assignment.

Performance with image width 1000 pixels:

The performance of this model is incredibly good in some ways but not for this specific application. The accuracy is 87% which is fairly high but more importantly the precision for the non-defective nails is 100%, this means that the model never classifies a nail that is non-defective as defective. This is great because it means that defective nails would not move past this process and would thus not be delivered to customers. The recall of 77% for the non-defective nails is fairly low which means that only 77% of all nails that are non-defective are predicted as non-defective, which results in a very wasteful process which could only be justified if the cost of delivering a defective product is very high.

Classification report:

	precision	recall	f1-score	support
0 (bad)	0.78	1.00	0.88	18
1 (good)	1.00	0.77	0.87	22
avg / total	0.90	0.88	0.87	40

Confusion matrix:

		Predicted	**Predicted**
		0	**1**
Actual	**0**	TN = 18	FP = 0
Actual	**1**	FN = 5	TP = 17

Accuracy: 0.875

Performance with image width 500 pixels:

The model that was trained on lower quality images performs significantly worse and is only worth considering if the latency of the good model is not high enough to classify every unit that passes the inspection point and if more value can be gained from increasing the sample size of units that are classified.

Classification report:

	precision	recall	f1-score	support
0 (bad)	0.75	0.83	0.79	18
1 (good)	0.85	0.77	0.81	22

avg / total	0.81	0.80	0.80	40

Confusion matrix:

		Predicted	**Predicted**
		0	**1**
Actual	**0**	TN = 15	FP = 3
Actual	**1**	FN = 5	TP = 17

Accuracy: 0.8

Questions:

How many images are required to build an accurate model?

For the task of building an anomaly detection system it would take a few thousand images taken on the same set up with the same conditions to achieve the optimal performance.

The images need to be taken in such as way that they cover the translational and rotational feature space as much and as uniformly as possible. This can be done manually by taking more pictures where the distribution of the orientation and translation of the nail in the image can be described as a uniform. This can also be done digitally using data augmentation processes.

Where do you see the main challenge in building a model like the one we asked here?

The images provided have a high amount of translational and rotational variability in the position and orientation of the nail on the table. And since the position and orientation don't relate to the quality of the nail most translational and rotational invariant machine learning models use this information to try to make predictions or to try to identify patterns. This variability distracts the model from learning the features associated with the nail itself such as the length, width, straightness and other geometric features.

Training a neural network like a Convolutional Variational Auto Encoder, which would be an ideal algorithm to explore for an image clustering task, requires a lot of images.

What would you do if you had more time to improve the model?

I would move away from the supervised approach until a use case arises that the supervised model is not suited for. For classification and anomaly detection I would focus strictly on the supervised learning approach. I would do this by experimenting with algorithms that are even better suited for image classification like CNNs and by focussing on improving the performance less computationally expensive lower latency algorithms such as gradient boosting classifiers.

What problems might occur if this solution would be deployed to a factory that requires automatic nails quality assurance?

Manufacturing facilities and machinery are not always the safest or cleanest of places. There are many factors that can make the installation difficult:

1. There is no easily accessible and reliable power source close to the ideal place to mount the system.
2. The ideal place to mount the system could be inaccessible or difficult to access or would require modification.
3. The system would interfere with the current manufacturing process.
4. The ideal place to mount the system would require a custom built rig.
5. There is no reliable internet connection.
6. There is no engineer or relevant technical personnel available that can assist with the installation if need be.
7. The light could be very dim and volatile.
8. The air quality could be contaminated by smoke, steam, lubrication and/or cooling fluids which would obscure the view.
9. The lens of the camera could be contaminated by moke, steam, lubrication and/or cooling fluids.
10. The temperature in the ideal place to mount the system could be to high for the system.
11. Receiving high quality training data pre installation can be difficult.
12. The throughput velocity of production is higher than the speed at which predictions can be made and defective nails can be removed.
13. There is no point in the process where the nails are standing still long enough to take pictures.

Once the system is installed one of the main sources of problems is that real world conditions are often impossible to predict and difficult to control. There are many factors that can make the maintaining the system difficult to maintain throughout the system life cycle:

1. The parameters of the manufacturing line could change without these changes being communicated to the system.
2. The system is deployed on site and can in some ways only be debugged in person.
3. The lens of the camera has to be cleaned routinely in a specific way.
4. There is no reliable internet connection.
5. There is no easily accessible and reliable power source close to the ideal place to mount the system.

6. The place where the system was mounted makes it difficult to maintain.

The second source of problems is the system itself. Since the defect prediction and localization system consists of multiple physical components powered by multiple software components these represent multiple possible points of failure.

Determining the value of a machine vision system:

To determine the value that a defect prediction or defect localization system delivers to the manufacturer various factors need to be considered.

N	=	Nd + Nn	=	Number of units produced per day
Smi	=	Sample size for manual inspection		
Cmqc	=	Cost of manual quality control per unit		
Cp	=	Cost of production per unit		
Cdi	=	Cost of disposal per unit		
Cr	=	Cost of repair per unit		
Cd	=	Cost of delivering a defective unit		
P	=	Precision (for label = non-defective) of the model		
R	=	Recall (for label = non-defective) of the model		
Nd	=	Number of units predicted as defective per day		
Nnd	=	Number of units predicted as non-defective per day		
Dr	=	defect rate (~0.01)		

These variables can then be used to determine the value gained per day. For the example below we assume that there is no manual inspection process and if are classified as defective they get disposed of.

Smi	=	0
Cmqc	=	0
Cr	=	0

With the defect prediction system:

Cadm = Cost of automatically disposing units that are misclassified per day

= Nd * (1 - R) * (Cdi + Cp)

Cdd = Cost of delivering defective units per day

= Nnd * (1 - P) * (Cd + Cp)

Without the defect prediction system:

Cdd1 = Cost of delivering defective units per day

= N * Dr * (Cd + Cp)

Value gained per day = (Cdd1) - (Cdd + Cadm)