Available online at www.prace-ri.eu

Partnership for Advanced Computing in Europe

SHAPE Project Vision-e: Deep Tile Inspection

Rudy Melli [a]* Eric Pascolo[b]

[a]*Vision-e S.r.l. .*
[b]*CINECA*

Abstract

Vision-e is an Italian company focused on machine vision application. One of the main market is the ceramic tile where inspection system is used to automatically check the quality of the product. The fourth industry revolution tends to create factories with few human controllers and need automatic systems without configuration or supervision. This project goes in that direction trying to increase the performance of actual machine vision systems we created, in particular the clustering phase that is crucial for a good inspection. We investigated Deep Learning technologies applied to clustering tiles using the DeepCluster approach and the good results we get, show us that this is the right way to reach the goal and to continue, starting from a study on the choice of input data.

1. Introduction

Vision-e S.r.l. born as a Spin-Off of Università degli Studi di Modena e Reggio Emilia. The founders come from the ImageLab staff research group, the research laboratory in Computer Vision of the same University; collaboration with ImageLab still continue on several projects that makes the team steadily up to date, able to provide a service that update and technological innovate itself with new research discovers.

The company is focused on study, design and develop computer vision systems and algorithms for custom industrial application of quality inspection. The team have an academic approach to the problem solving, mixed with the rational industrial point of view acquired on the field during last years. This kind of knowledge is poorly present in national context and, often, is limited only to academic research groups.

In the last 10 years we study and develop a tile's inspection system software based on computer vision to detect surface defects of tiles on production line called *VEInspector*.

One of the main problems, on tiles inspection systems, is the huge variety of ceramic tiles that change in terms of colours, glossy, texture, structures and each single product had dozens or hundreds of different *surfaces* (the shape above the ceramics). Moreover, the minimum defects to found is often very little and could be hidden by the graphics, so it is mandatory to train a model of each surface in order to discriminate between real defects and graphics.

For this kind of applications, the operation could be summarized with this list:

1. Localization. Detection of the border of the tile to fix the working ROI.
2. Training. This phase is done for each type of tile (product) before to be inspected and create a model for each one. Usually the system acquires a batch of tile images depending by the texturing of the graphics and by the number of surfaces.
3. Matching. This operation is the first one during the normal inspection and is necessary to identify which surface are captured and use corresponding model.

* Corresponding author. *E-mail address*: melli@vision-e.it

4. Inspection. This is the core of inspection system and is based mainly on the model chosen by the clustering, if the choose is not correct the inspection not work properly and could detect false positive and false negative

To increase productivity and reduce cost, the ceramic tile's factories requires to inspection systems to be almost automatic without human support. The tiles arrive on a random surface order then the system must recognize the similar surfaces in order to create the corresponding model or starting the inspection.

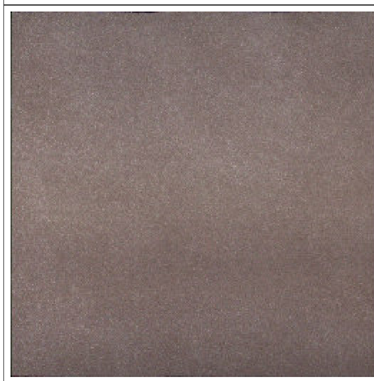
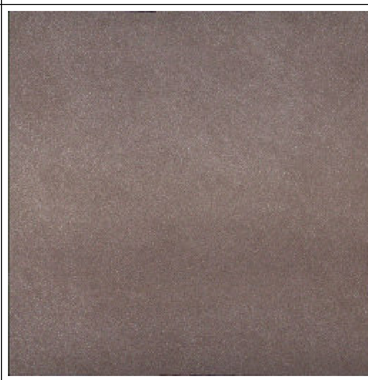
 Tile 1 - Surface A	 Tile 1 - Surface A	 Tile 1 - Surface B
 Tile 2 - Surface A	 Tile 2 - Surface A	 Tile 2 - Surface B

Table 1: Example of tile shapes. In the first and second column 2 different images of same surface, in the third a different surface.

In Table 1 are shown 2 examples, row 1 and row2, of different tile shapes. In the first and second column are presented two images of different tiles but belonging to the same surface, whereas in the third column is present a different surface. The difference between tile of the same surface could be considerable due the method used to create the different surface graphics, working process and raw materials.
In the second example of Figure 1 both surfaces are very similar and, in these cases, the automatic clustering is critical, because is difficult to separate. In these cases, the biggest challenge is the training process.
Clustering difficulties increase because often the inspection system doesn't know how many surfaces the product has.

Current inspection systems are based on a computer vision classical approach [2],[3],[4] that decrease seriously performance in case of textured tiles. The clustering is often based on appearance features and comparison metric use pattern matching or
Machine learning was one of the first way to inspect textured tiles [5] in case of tiny variations, it cannot group correctly the surfaces and need human action to verify and correct the tile's surfaces classification, for this reason is called "semi-automatic".

Within this PRACE SHAPE project, the focus was the investigation of the use of neural network technologies in order to find an unsupervised clustering algorithm that can automatically group tile's surfaces.

2. Computational platform

To achieve our objectives, we used D.A.V.I.D.E. HPC cluster operated by CINECA [1]. D.A.V.I.D.E. is a supercomputer based on an enhanced version of the CPU called POWER8+ that includes a new high-speed bus (NVLink) used by the latest NVIDIA GPU based on Pascal Architecture: Tesla P100.

Nvidia Tesla P100 was built to deliver high performance for the most demanding computing applications and it's the top-level GPU card on server side for neural networks application. Peak performance can be summarized as follows: 5.3 TFlops of double precision floating point (FP64); 10.6 TFlops of single precision (FP32); 21.2 TFlops of half-precision (FP16).

Each POWER8-NVLink CPU is connected to 2 NVIDIA Pascal P100 via PCIe to provide power and management communication, while the data movement takes advantage of the NVLink 1.0 bus with a bi-bandwidth of 80 GB/s, between both CPU-GPU and GPU-GPU

The compute nodes derive from the OpenPOWER system designed with the codename Garrison. Each node will host two IBM POWER8+ with NVLink and 4 Pascal P100 with the intra node communication layout optimized for best performance. The original design of the server is air cooled, while the implementation for the Pilot system will use direct liquid cooling for CPUs and GPUs. A prototype version of the liquid cooling is shown in *Figure 2*. Each compute node has a peak performance of 22 TFlops in double precision and an estimated power consumption of 2 kW

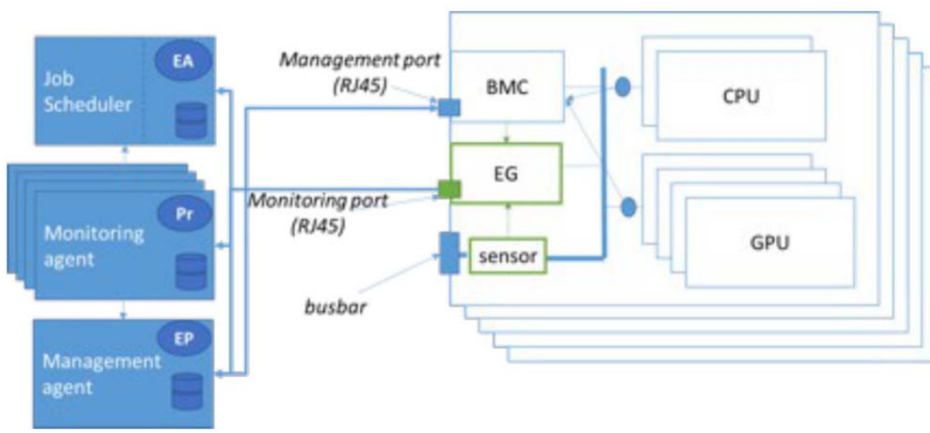


Figure 1: Main functionalities of D.A.V.I.D.E. power aware usage

3. Background

Data clustering is a basic problem in many areas, such as machine learning, pattern recognition, computer vision, data compression. The goal of clustering is to categorize similar data into one cluster based on some similarity measures (e.g., Euclidean distance).

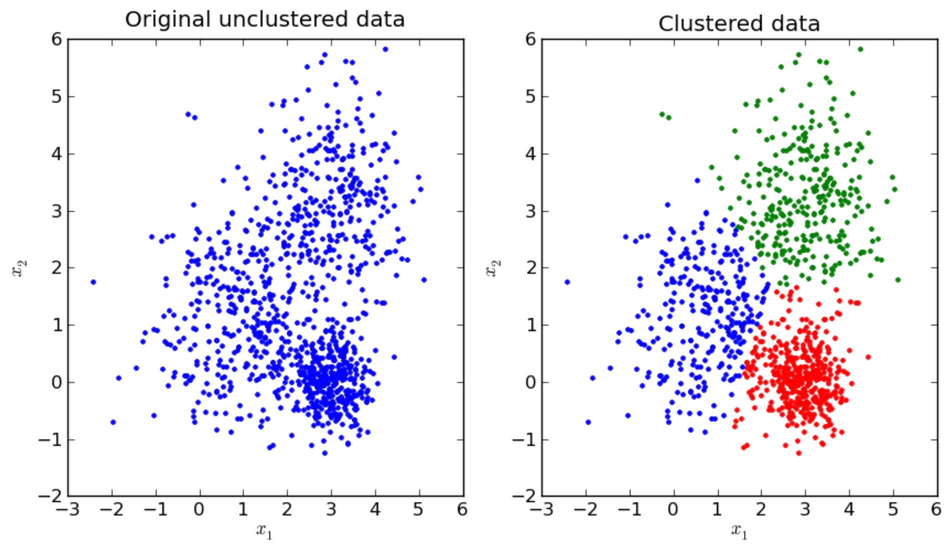


Figure 2: Example of clustering. Source: https://i.stack.imgur.com/cIDB3.png

Clustering of images with deep learning is an almost novel approach, due to the age of deep learning, but yet with a big effort in academic research. The related works can be summarized into three groups:

- Unsupervised learning of features: Yang et al. [6] iteratively learn Convnet (convolutional neural network) features and clusters with a recurrent framework. Their approach works well with small dataset but increase complexity and performance scaling the number of images. Bojanowski and Joulin [7] learn visual features on a large dataset with a loss that attempts to preserve the information flowing through the network
- Self-supervised learning: Doersch et al. [8] use the prediction of the relative position of patches in an image as a pretext task. Noroozi and Favaro [9] train a network to rearrange shuffled patches spatially. These approaches are usually domain dependent, requiring expert knowledge to carefully design a pretext task that may lead to transferable features
- Generative model: Donahue et al. [10] and Dumoulin et al. [11] have shown that using a GAN (Generative Adversarial Network) with an encoder, results in visual features that are pretty much competitive for clustering

In 2018 Facebook AI Research team present a clustering method called *DeepCluster* [12], actually is the state of the art in this field of research. It obtains useful general-purpose visual features with a clustering framework. DeepCluster approach is robust to a change of architecture and maintains state-of-the-art performance when trained also on uncured data distribution. They use ImageNet [13] as a training set for unsupervised models for its particular image distribution inherited from its use for a fine-grained image classification challenge: it is composed of well-balanced classes and contains a wide variety of dog breeds for example.
The method works with standard clustering algorithm, like k-means and others, that iteratively groups and uses the subsequent assignments as supervision to update the weights of the network. The good performance of random convolutional networks is intimately tied to their convolutional structure which gives a strong prior on the input signal. The idea of DeepCluster is to exploit this weak signal to bootstrap the discriminative power of a convolutional neural network. They cluster the output of the convnet and use the subsequent cluster assignments as "pseudo-labels". The method approach iteratively learns the features and groups them.

DeepCluster is implemented with a standard AlexNet architecture with five convolutional layers and three fully connected layers, removed the Local Response Normalization layers and used batch normalization. A linear transformation based on Sobel filters is used to remove colour and increase local contrast.

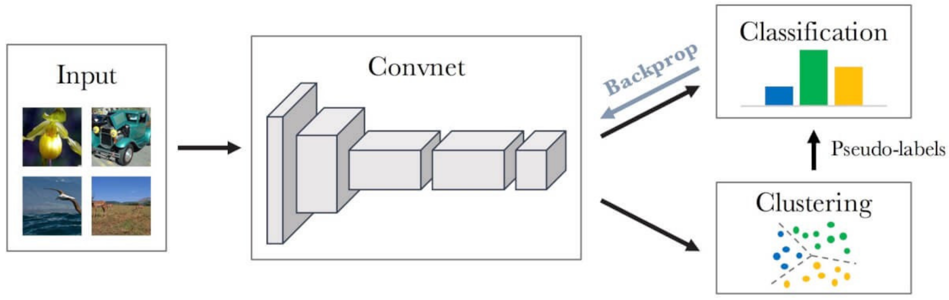


Figure 3: DeepCluster

4. Methods

In this works we investigate the DeepCluster approach applied to our clustering problem. The boundary conditions are different because the images we use contains only the tile, without background, and the difference between the surfaces could be slight. Another problem could be the little number of images for each surface that we can use for training. This is due to factory production constraints; is not always possible to collect a lot of sample for each tile, because the production's batch could change after few thousands of pieces.
For these reasons was important to investigate the behaviour of the state of the art deep clustering network and compare with previous method.

We use three different datasets of three different tiles, choosing the hardest to clustering correctly with previous method, each of it with several surfaces.

Tile	# surfaces	Total # images	Average # images per surface
TileA	10	5020	502
TileB	25	6540	261,6
TileC	48	5708	118,92

Table 2: Dataset used in our research

We implement DeepCluster in our workstations (Xeon, SSD, 32GB RAM, Nvidia GTX 1080TI) and on HPC D.A.V.I.D.E..
On D.A.V.I.D.E. we slightly change the architecture of the training process using the multi-gpu capabilities of the machine with the power of four Nvidia P100. This was done with data parallelism of the model, splitting the input images set into multiple smaller batches and run the computation for each of the smaller batches in parallel. This is not possible on workstation that have only one GPU. This approach does not affect the calculation and the numeric results that, in both architectures (workstation and D.A.V.I.D.E.), are absolutely the same.

The tests started using the pre-trained model provided by authors and refining the training with our datasets.
Each dataset is randomly divided in two sets: 80% for training and 20% for evaluation.

In Figure 4 are present the results of loss score during training process after 100 iterations, lower values is better. The TileA dataset performs better compared to TileB and TileC due to less number of surfaces and higher number of images per surface. TileB dataset converge less than TileA with slightly higher value. TileC dataset loss score, remain high and unstable, maybe due to the high number of surfaces and few images to use for training.

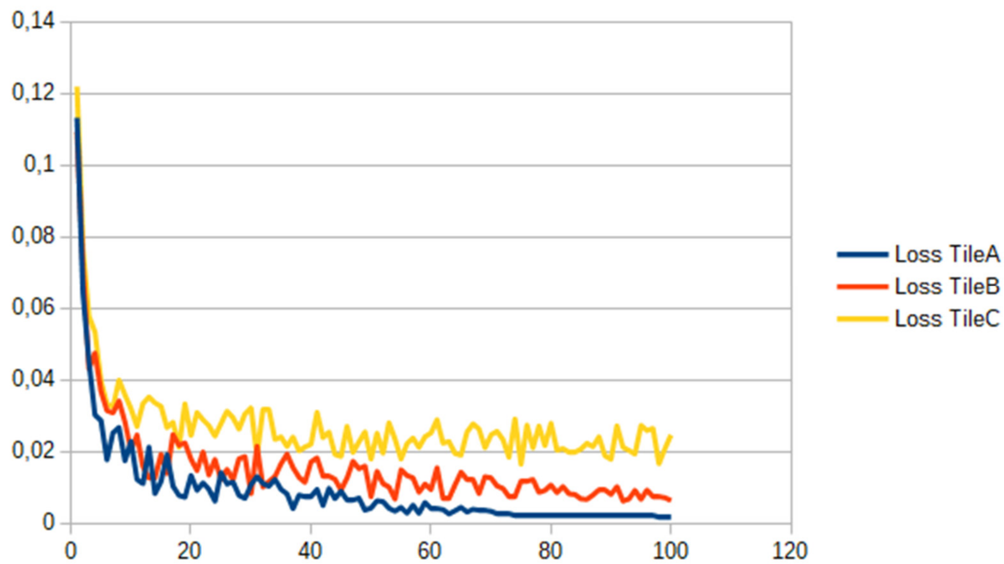


Figure 4: Loss function during training

The great improvement we had from using HPC server was the **training time** where we have a reduction factor of about 3X that has allowed us to speed up the analysis of the large amount of data.

	Average training time (sec/iteration)	Factor
Local workstation	278	1X
HPC D.A.V.I.D.E.	86	**3.2X**

Table 3: HPC vs workstation - time comparison on training

Another useful advantage of HPC is the task's scheduling feature that allow us to program multiple instances of training with different configurations decreasing significantly the time need to manage the tests.

5. Results comparison

In order to evaluate if DeepCluster could be better than previous method based on traditional computer vision, we compare the results. Cluster evaluation have not a single solution and in our case the dominion of two methods avoid to use metrics based on distance. For the comparison we choose to use *purity* metric [14] that is a simple and transparent evaluation measure, greater value of *purity* corresponds to better clustering. All tests were performed on D.A.V.I.D.E.. Results in Table 4 show that DeepCluster works better for dataset TileA and TileB but perform worst for TileC may be due to the little number of images for each surface. This is a great result because it demonstrate the good performance of a deep learning approach that could be increased working on input data.

	Previous method	DeepCluster
TileA	0.91	**0.97**
TileB	0.87	**0.90**
TileC	**0.80**	0.72

Table 4: Comparison results

6. Conclusions

This project demonstrate that deep clustering could work better if the number of input data for training is enough and can produce a completely automatic inspection system. However, the final *purity* score for some cases (B and C) is not yet over 0.95 that is our internal quality threshold to determine if clustering could be considered good enough to be used *unsupervised*. But this is only the first investigation and we want to continue to study on this technology in order to improve results and reach the goal of an *unsupervised inspection* system, a revolutionary product that does not need skilled operators to setup and that increase its performance. This will raise our market share and quality in the service of ceramic tile production. For this purpose, HPC will be a great partner in order to speed up significantly the research and reduce the time-to-market.

References

[1] WA Ahmad, A Bartolini, F Beneventi, et al., "Design of an energy aware petaflops class high performance cluster based on power architecture", 2017 IEEE International Parallel and Distributed Processing Symposium Workshops

[2] Xianghua Xie, "A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques", Electronic Letters on Computer Vision and Image Analysis 7(3):1-22, 2008

[3] ML Smith, RJ Stamp, "Automated inspection of textured ceramic tiles", Computers in Industry, 2000 – Elsevier

[4] X Xie, M Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007

[5] Ataollah Ebrahimzadeh, Mahdi Hossienzadeh, "An efficient system for automatic sorting of the ceramic tiles", 6th International Conference on Digital Content, Multimedia Technology and its Applications2010

[6] Jianwei Yang, Devi Parikh, Dhruv Batra, "Joint Unsupervised Learning of Deep Representations and Image Clusters", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)2016

[7] Piotr Bojanowski, Armand Joulin, "Unsupervised Learning by Predicting Noise", ICML'17 Proceedings of the 34th International Conference on Machine Learning, Volume 70, Pages 517-526

[8] Carl Doersch, Abhinav Gupta, Alexei A. Efros, "Unsupervised Visual Representation Learning by Context Prediction", 2015 IEEE International Conference on Computer Vision (ICCV)2015

[9] Mehdi Noroozi, Paolo Favaro, "Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles", ECCV 2016

[10] Jeff Donahue, Philipp Krähenbühl, Trevor Darrell, "Adversarial Feature Learning", ICLR 2016

[11] Vincent Dumoulin, Ishmael Belghazi, et al., "Adversarially Learned Inference", ICLR 2017

[12] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. "Deep Clustering for Unsupervised Learning of Visual Features." Proc. ECCV (2018)

[13] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., "Imagenet: A large-scale hierarchical image database". CVPR. (2009)

[14] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze,"Introduction to Information Retrieval", Cambridge University Press. 2008, cap 16.3, pag. 356

Acknowledgements

This work was financially supported by the PRACE project funded in part by the EU's Horizon 2020 Research and Innovation programme (2014-2020) under grant agreement 730913.