COPEN-13

13th International Conference on Precision, Micro, Meso and Nano Engineering

Machine vision-based pre-machining inspection of fixtured parts

Madhav Kumar, Mohit Law*

Machine Tool Dynamics Laboratory, Department of Mechanical Engineering, Indian Institute of Technology Kanpur, Kanpur - 208016, India

* Corresponding author e-*mail address:* mlaw@iitk.ac.in

Abstract: Mass production of parts on modern CNC machines is only possible if the part is correctly located, supported, and secured in a fixture. Though fixturing systems have become increasingly automated, loading and unloading of the part before and after machining is still usually done manually. Though fixtures are designed to be foolproof, it is possible that due to operator fatigue and/or unintended negligence, the part is incorrectly loaded. This can cause an accident. To avoid such situations, we propose a machine vision-based solution in which images of the loaded parts are used to classify if the part is correctly loaded or not. We train the model with synthetic data that includes many combinations of faulty loading scenarios and demonstrate that the model can correctly classify a fault with up to 100% accuracy. Such systems can be easily integrated within CNC machines to only allow machining of parts that are classified as being loaded correctly.

Keywords: Machine vision, fixture, classification, machining, support vector machine.

1. Introduction

Mass production of parts on modern CNC machines is only possible if the part is correctly located, supported, and secured in a fixture. Since CNC machines start machining from the same reference point each time, they require the part to be located correctly. Though there exist robust guidelines for the design and foolproof use of fixtures to ensure that the part is loaded correctly prior to machining [1], and even though fixturing systems have become increasingly automated, with clamps and supports being integrated with sensors that detect if the part is present and secured properly [2], loading and unloading of the part before and after machining is still usually done manually. Since mass production of parts could require operators to repeatedly load and unload parts, it is possible that due to operator fatigue and/or unintended negligence, the part is incorrectly loaded. This could result in accidents that may damage the part, the fixture, the tool, the machine, and even the operator. The easy fix would be to automate loading and unloading too with sensory interlocks such that machining will only begin when the part is correctly loaded. However, loading and unloading using robots or gantry systems is usually expensive and requires hardware-software interfacing capabilities beyond the capacity of most modern machine shops.

Since checks for parts being loaded correctly are necessary, this paper proposes a low-cost machine vision-based solution that uses images of the loaded parts to classify if the part is correctly loaded or not. Demonstrating the workings of this solution is the main new modest technical contribution of this paper.

Prior research on the use of machine vision in machine tools has focused on classifying tool wear [3], on inspecting machined surface quality [4], on measuring machine tool vibrations [5], and also on machine tool metrology [6]. Though the use of machine vision in machine tools is not new, our use of it to classify parts being correctly loaded certainly is.

The central idea in our proposed solution and its implementation is that a low-cost camera can be easily mounted within the machine's working area, and any one image of a part mounted correctly can then be used to classify if any other part of same design is correctly loaded or not by using a trained machine learning model capable of such classification. We demonstrate the workings of this solution using only synthetic data. We synthetically generate what a good image of the part is using Blender - an open source, 3D CG code [7] to generate an image dataset of parts that could have been mounted wrong. We also generate images with blur and inappropriate lighting – both of which could happen in real-world settings. For classification, we use two well-established supervised learning methods – one based on neural networks [8], and the other based on support vector machine [9].

The remainder of the paper is organized as follows. Section 2 describes what a correctly loaded part looks like. Section 3 discusses generation of synthetic data to be used to train a model that can classify a good, correctly loaded part from a bad, incorrectly loaded part. Section 4 discusses our implementation of the machine learning model, followed by the main results in Section 5. Main conclusions and outlook follow in Section 6. We take this opportunity to clarify here that although we demonstrate the solution with synthetic data, such solutions and systems can be easily integrated within CNC machines to only allow machining of parts that are classified as being loaded correctly.

2. What a good, correctly loaded part looks like

This section outlines what a good, correctly loaded part looks like. The part is rectangular with a non-symmetric profile that must be finish cut on it. The part is shown in Fig. 1. This part is representative of any part being loaded on any CNC machine. This part and its setup were generated in Blender. In this case the part is assumed to be mounted on a fixture that in turn is mounted on the table of a vertical machining centre, and the primary machining operations to be performed are contour milling. The profile shown in the part is what the part will look like before finish machining it. The rough machined profile is the same as the finish machined one with the only difference being the size and depth of the profile.

The part is supported on four pins, one or more of which can act as locators and supports. The part is clamped with four clamps (C1 – C4), and this arrangement to support, locate, and clamp arrests all six degrees of freedom of the part. The clamps though shown to be of the manual, mechanical type, could as well be of the hydraulic and/or pneumatic type. The configuration and orientation of the part shown in Fig. 1 is the correct loading case. Also shown in Fig. 1 is the fixed location of the camera mounted within the machine. There are also three light sources: L1 – L3. Two of these light sources are near two of the clamps and a third light source is near the camera.

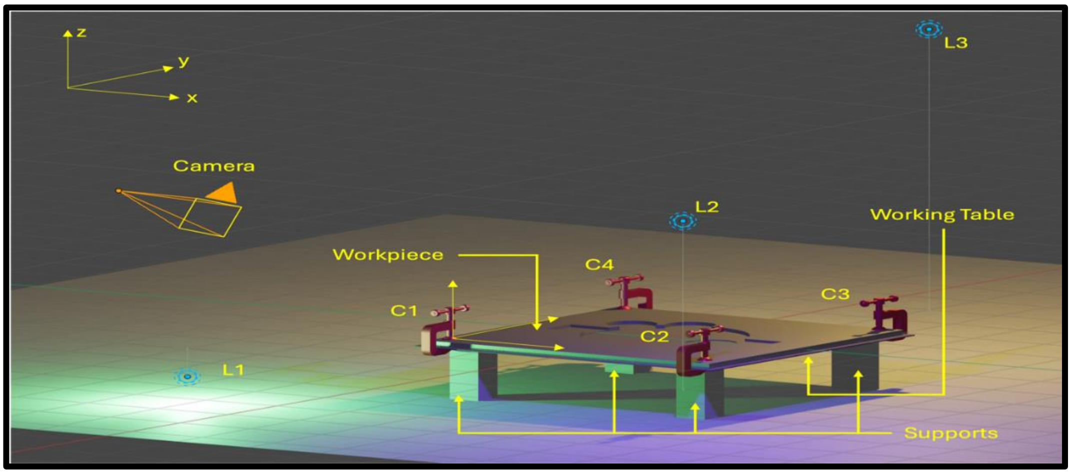

Fig. 1. The overall fixturing setup showing the camera, the workpiece, clamps- C1, C2, C3 and C4, the light sources- L1, L2 and L3 and the table on which the machining is to be done.

3. What bad, incorrectly loaded parts can look like

Since the machine learning model needs to be trained to distinguish a correct part from an incorrect one, this section discusses what bad, incorrectly loaded parts could look like. The first case could be that of the part missing altogether, i.e., the part not being loaded by the operator, but the cycle still being started. Other cases could include parts mounted in the wrong orientations, or the part being mounted flipped, or of one of the four clamps not being secured. A combination of these errors is also possible, such as the part being mounted in the wrong orientation and a clamp not being secured. Since there are many such potential cases, we generate a large synthetic data set of 196 images using Blender.

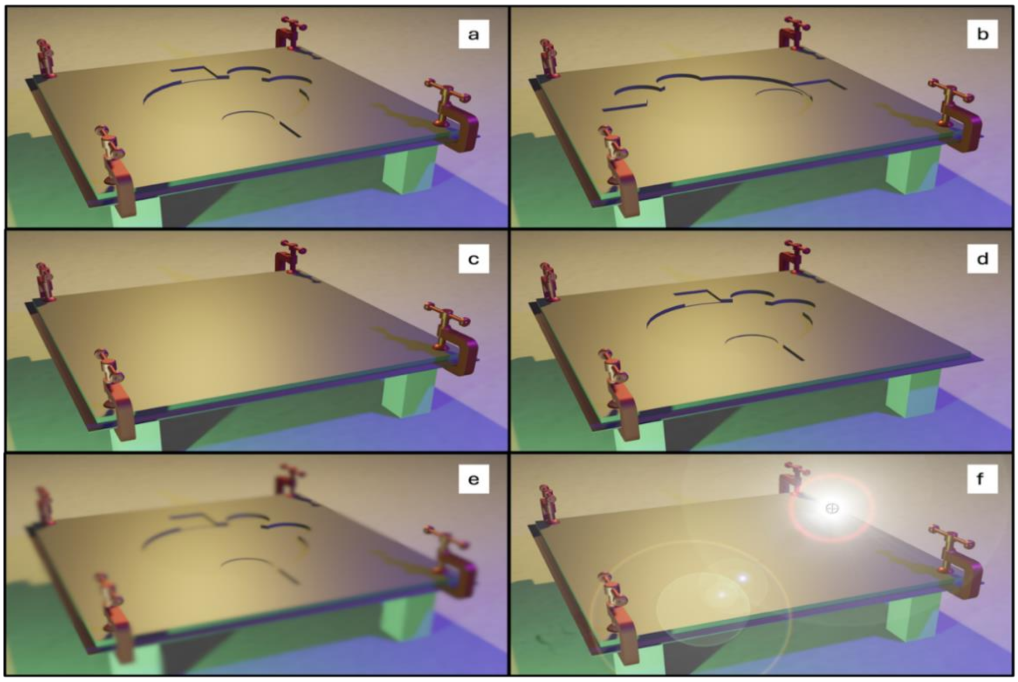

Fig. 2. Representative images as captured by the camera. (a) Shows the clear image of correct condition, (b) wrong orientation of the workpiece, (c) wrong side up, (d) missing clamp, (e) noisy image of correct condition, (f) light glare added to the image of a part with loaded with the wrong side up.

Moreover, since machine learning models work better when data sets are larger, we augment this data by adding Gaussian noise to some images. Noise in images could be from distortions due to shadows cast by the lights or due to glare by lights, or worse still when one or more of the lights is not functional, the image acquired could be dark. It is also possible that residual mist due to having used coolant during cutting distorts the acquired image. It is likely that due to the acquired image being distorted, that a correct loading of the part could be wrongly classified. We thus generate a total data set of 528 images, which includes 332 images that are distorted. Representative examples of correctly and incorrectly loaded parts, clear as well as distorted are shown in Fig. 2.

4. Methods of classifying correctly and incorrectly parts

The problem being dealt with in this paper is of the 'binary classification' type, i.e., classify a good part from bad. Input data is an image, and the output is the probability value for the two classes based on the input image. If this probability is greater or less than some pre-defined threshold value it is classified into one of the two groups. Else, it is declared 'not predictable'. Although it is probable, but the last case is hardly seen if the training data is good enough in terms of diversity and sample size. This paper discusses the use of supervised Machine Learning methods for the classification task.

We discuss the use of two commonly used classifiers: densely connected neural networks (NN), and support vector machines (SVM). We have also made use of a neural network-based transfer learning technique, which enhances the convergence rate of machine learning models significantly. We used an open-source pre-trained model, ImageNet [10], trained on large datasets of diverse categories of images. For the neural network-based model, we created a convolutional neural network (CNN) for binary classification consisting of a base model (transfer learning part), MobileNetV2 [11], trained on the 'ImageNet' dataset and some densely connected layers. SoftMax activation function was used in the last layer for getting prediction probabilities. Another category of supervised learning technique called Support Vector Machine (SVM) was also analysed. Transfer learning is used here to build the feature space which is used for training the SVM.

The total number of images in the dataset was 528 with equal number of images of both classes. The CNN was trained on 60% data and tested on the remaining 40%. Main results are presented next.

5. Main results

The efficiency of the binary classifier is assessed by the accuracy of the model on the test data. Accuracy in this context is a measure of how well the predictions from the machine learning model match reality, i.e., it is a measure of correct predictions to total predictions. In this case, since we use a supervised machine learning algorithm, we a priori know which image corresponds to a part that is correctly loaded, and which corresponds to a part that is incorrectly loaded. As such, accuracy for us is the metric by which we gauge how well the prediction of the classification of correctly or incorrectly loaded part matches the known condition of that part.

In the case of the dense neural network-based model, we observed the accuracy to be a function of the number of epochs used to train the network. In general, more epochs were observed to improve accuracy. We achieved an accuracy of 98% with 10 epochs, 99% with 20 epochs, and 100% with 30 epochs. Since the computation effort with 30 epochs is not significantly larger than with 10 epochs, and since the accuracy is 100% with 30 epochs, we recommend that the neural network be used with more epochs.

For the SVM based binary classifier, and a linear kernel, using 60% of the data for training, resulted in an accuracy of 100%, whereas if only a meagre 2% of the data (that corresponds to only nine images) was used for training, then the accuracy was observed to drop to 88%. This shows that SVM produces very good results for small to normal sized datasets compared to the dimension of individual data points (here dimension refers to the total number of pixels making up the image). The SVM also trains significantly faster than a neural network.

To demonstrate easy deployment of our proposed solution, we have developed a windows-based application with an interactive GUI. The application can be used to train and save a model based on a labelled image dataset. This dataset comprises of two folders, one with images of correctly loaded parts and the other containing images of incorrectly loaded parts. The saved model is standalone, meaning it can be trained and used on two different computing machines with completely different environments. This model can then be loaded to classify the images in real-time with an average computation time of 110 millisecond on MacBook Pro M1 2020. The GUI of the application is displayed in Fig. 3 which we named 'Vargikaran', which is the Hindi-language word for classification. Fig. 3 also shows the results from the classification task. On the left is an example of a correctly loaded part, and the classification task identifies it as so. On the right is an example of an incorrectly loaded part, and in this case too the classifier identifies the wrongly loaded part correctly.

Fig. 3. Screenshot of the application in action. The objects are classified, and the class is displayed.

6. Conclusions and outlook

This paper presented a new way to inspect if a part has been correctly loaded or not on a fixture by using machine vision methods. The solution is premised on integrating a low-cost camera within the machine's working area that takes images of the parts loaded by the operator to classify if such parts are loaded correctly. We successfully demonstrated that machine vision can successfully classify correct parts from incorrect ones with an accuracy of up to 100%. Though we trained and tested the algorithms with synthetically generated data that is representative of real parts to be fixtured on real machines, the method is robust enough to be deployable with real data. We believe that such solutions can be easily integrated within CNC machines to only allow machining of parts that are classified as being loaded correctly, and such integration forms part of the planned future work. We also believe that such solutions are not limited only to machining but can be easily implemented in other manufacturing applications such as joining processes that also require parts to be oriented and held correctly before welding.

Though we demonstrate the workings of the proposed solution with synthetic data, real-world implementation will likely need addressing additional challenges related to appropriate mounting of cameras within the workspace in machines such that the camera sees what it must and such that its line of sight is not obstructed by coolant splashes and flying chips. Real-world implementation will also need to address interfacing the camera with an edge device that can remotely transfer data to a server for the classification task. Successful implementation will also require potential costs to be kept low.

References

[1] Grant, H. E., 1967, Jigs and Fixtures: Non-Standard Clamping Devices. McGraw-Hill.

[2] Moehring, H. C., Wiederkehr, P., Gonzalo, O., and Kolar, P., 2018, “Intelligent Fixtures for the Manufacturing of Low Rigidity Components,” Springer, Cham.

[3] Dutta, S., et al, 2013, “Application of Digital Image Processing in Tool Condition Monitoring: A Review.” CIRP Journal of Manufacturing Science and Technology, 6/3: 212–232

[4] Singh, S. A., Kumar, A. S., and Desai, K. A., 2023, “Vision-Based System for Automated Image Dataset Labelling and Dimension Measurements on Shop Floor,” Measurement, 216, p. 112980.

[5] Law, M., 2024, “Vision-based modal analysis of machine tool systems: progress and prospects,” Journal of Flow Visualization and Image Processing.

[6] Mori, K., Kono, D., and Matsubara, A., 2023, “Vision-based Volumetric Displacement Measurement with a Self-illuminating Target,” *CIRP Annals*, Vol. 72, No. 1, pp. 305-308.

[7] Community, B.O., 2018. Blender - a 3D modelling and rendering package, Stichting Blender Foundation, Amsterdam. Available at: http://www.blender.org.

[8] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., 1998, “Gradient-Based Learning Applied to Document Recognition,” *Proceedings of the IEEE*, 86(11), pp. 2278-2324.

[9] Cortes, C., and Vapnik, V., 1995, “Support-Vector Networks, *Machine Learning*,” 20(3), pp. 273-297.

[10] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., 2009, "ImageNet: A Large-scale Hierarchical Image Database," IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255.

[11] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., 2018, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4510-4520.