COMPUTER VISUAL INSPECTION OF PEAR QUALITY

Mykhailo Ilchuk, PhD Student; Andrii Stadnyk, PhD Student

Lviv Polytechnic National University, Ukraine; e-mail: astadnyk21@gmail.com

https://doi.org/

Abstract. A brief description of the basic stages of image processing is given to pay attention to the segmentation stage as a possible way to improve efficiency in decision-making. The main characteristics of the presented model are visual signs, such as color, shape, the presence of a stem, and others. Due to the different approaches in image processing, a high level of truthfulness is achieved, which is expressed in the percentage ratio of the accuracy of decision-making and varies in the range from 90 to 96%. Therefore, the results obtained in this work make it possible to automate the process of visual inspection with the prospect of increasing the speed and quality of product sales for the consumer.

Key words: Quality control, Computer vision, Automation

1. Introduction

The task of visual inspection to recognize objects and assess their quality is one of the most important processes in the processing and food industries. Given the requirements of the customer, it is mandatory to ensure the highest quality of the product. The issue of inspecting objects to detect defects such as color, scratches, and cracks, or checking surfaces for proper coating is related to visual quality control [1]. One alternative adopted by many industries to remain competitive is to promote lean manufacturing, in which these techniques can work synergistically to create a streamlined, high-quality system that produces finished products at the pace of consumer demand with little or no waste. Unfortunately, criticizing the available data, which suggests that several organizational factors can prevent or hinder the implementation of lean manufacturing methods among manufacturing enterprises. Another alternative is to give the computer the ability to automatically inspect and recognize objects. In our opinion, the use of specialized software together with other mechanisms, such as cameras, sensors, and the involvement of highly qualified specialists, allow us to obtain a powerful tool for automatic and rapid product quality control. Such automation provides an opportunity to reduce cost and speed up production with a high level of accuracy in establishing product quality. This technology can play an important role in fruit inspection. An important condition here is the observance of a non-invasive method, as well as taking into account the fact that its quality can be almost accurately indicated by the visual state.

Computer vision (CV) [2] deals with the modeling and reproduction of human vision with the help of software and hardware. It is the basis for creating artificial systems for extracting information from images. Since by analogy with a person who perceives more than 70% of the information of the surrounding world precisely through vision, such a model makes it possible to achieve significant success. The main task is to correctly interpret the information from the image we receive.

2. Drawbacks

Visual inspection remains one of the most important and fastest methods of non-invasive inspection of products and their classification by quality level during production and distribution to the consumer. This is usually done by the qualified operator whose task is to quickly make a decision and establish the fact of the presence of a visual defect. This is especially relevant in the food industry. This method has an obvious problem, namely the human factor. Since it is impossible not to take into account the operator's fatigue factor, and his subjectivity when choosing the acceptable level of product deviation.

Therefore, numerous studies have been developed in the field of computer vision. Nevertheless, still is absent a standardized method that can be proposed to evaluate the quality of different types of objects. Special characteristics of the object require the setting of the computer vision system; it involves an exhaustive research process, not just the purchase of expensive equipment to achieve better system performance and obtain better input data.

3. Goal

The goal of the study is to develop an automated model for visual inspection of product (pear) quality, with more than 90% correct conclusions on its condition.

4. Computer vision

The task is to present a brief description of the main stages and nowadays trends of computer vision systems with their critical analysis [3].

In application, the considered technology allows the automation and supplementation of human vision, creating many options for application. Due to advances in artificial intelligence and innovations in deep learning and neural networks, the field has been able to take great leaps recently and surpass humans in some tasks related to object detection and labeling. Before the advent of deep learning, the tasks that computer vision could perform were quite limited and required a lot of manual coding and effort on the part of developers and human operators [4].

Machine learning provided a different approach to solving computer vision problems [5-6]. Thanks to machine learning, developers no longer have to manually code each rule into their vision applications. Instead, they program "features," smaller applications that could detect specific patterns in images by applying statistical learning algorithms such as linear regression, logistic regression, decision trees, or support vector machines (SVM) to detect patterns, classify images, and detect objects. To create a satisfactory deep-learning algorithm, we gather a certain amount of labeled training data and tune the parameters such as the type and number of layers of neural networks. Compared to previous types of machine learning, deep learning is both easier and faster to develop and deploy.

Most current computer vision applications such as cancer detection, self-driving cars, and facial recognition utilize deep learning [7]. Deep learning and neural networks have moved from the conceptual realm into practical applications thanks to the availability and advances in hardware and cloud computing resources.

A quintessential example of transportation is the company Tesla's technology of manufacturing electric self-driving cars that rely solely on cameras powered by computer vision models. Computer vision enables self-driving cars to make sense of their surroundings. Computer vision also plays an important role in facial recognition applications, the technology that enables computers to match images of people's faces to their identities. CV algorithms detect facial features in images and compare them to databases of facial profiles. CV also plays an important role in augmented and mixed reality [8], the technology that enables computing devices such as smartphones, tablets, and smart glasses to overlay and embed virtual objects on real-world imagery. Using computer vision, AR hardware detects objects in the real world to determine the locations on a device's display to place the virtual object. CV enhances health tech. Its algorithms can help automate tasks such as detecting cancerous moles in skin images or finding symptoms in x-ray and MRI scans. We can see CV revolutionizing the retail space, such as the Amazon Go program, which introduces checkout-free shopping using smart sensors.

Automated computer visual control (ACVC) relies on the CV to capture visual information through cameras [9]. As in most industries, automation is useful for visual inspection. CV works as well with visual inspection systems as it does with others. First, provide the algorithm with a sample of a well-manufactured product. Once implemented, the system verifies each manufactured product. The system detects defects by capturing the product's image from multiple angles and comparing it with the pictures of the well-manufactured sample fed to the algorithm during setup.

The Edge Tracking method and anomaly detection were also important integrated parts of computer vision in tasks of visual control [10]. Anomalies are events that differ from the norm, occur infrequently, and don't fit into the rest of the "pattern". The motivation behind using anomaly detection is as follows. Quality Assurance needs to be automated to deal with variability (mass customization of products). To guarantee high quality, it is necessary to identify various quality problems. Human visual inspection does not guarantee reliable inspection for continuously changing products. Advances in technologies (both hardware and software) have decreased the cost of anomaly detection and made it affordable even for small businesses.

The edge detection algorithms are composed of 5 steps [11]: Noise reduction; Gradient calculation; Non-maximum suppression; Double threshold; Edge Tracking by Hysteresis.

CV has a lot to offer in terms of facilitating practical applications [12]. For practitioners or even those who entertain themselves with deep learning, it is very important to keep abreast of the latest developments in the field and stay up-to-date with the latest trends. Considering the current state of CV, several main trends can be identified.

The first one is *Resource-Efficient Model*. The main reason for its implementation is that the most modern models are often difficult to run offline on tiny devices such as mobile phones, Raspberry Pi, and other microprocessors [13]. And more complex models are inherent in the significant delay (i.e., the time it takes for the model to perform a direct path) and the considerable impact on the cost of infrastructure.

Therefore, sparse training refers to injecting zeros into the matrices used to train neural networks. This can be done because not all dimensions interact with others, or would be significant. Although performance might take a hit, resulting in a major reduction in the number of multiplications, reducing the time to train the network.

One closely related technique is pruning, where you discard network parameters that are below a certain threshold (other criteria exist as well). Using quantization in Deep Learning, to lower the precision (FP16, INT8) of models to reduce their size. With Quantization-aware Training (QAT), you can compensate for the loss of information caused by reduced accuracy. Pruning plus quantization can be the better approach. Training a high-performing teacher model and then distilling its "knowledge" by training another smaller student model to match the labels yielded by the teacher.

The next trend is Self-supervised Learning.

Self-supervised learning doesn't use any ground-truth labels but pretext tasks [14]. Then, consuming a portion of the unlabeled data set, we ask the model to learn the data set. Compared to Supervised Learning, where a humongous amount of labeled data is needed to push performance, labeled data is costly to prepare and can be biased as well, duration of training time is high for such a big data regime. It has the following features: Asking a model to be invariant to different views of the same image. Intuitively, the model learns the content that makes two images visually different i.e. a cat and a mountain. Preparing an unlabeled dataset makes the way cheaper. SEER (a self-supervised model) performs better than supervised learning counterparts in object detection and semantic segmentation in CV.

Self-supervised learning requires a big data regime to perform real-world tasks such as image classification. Therefore, it is quite expensive.

Another important trend is Robust Vision Models. They have been adopted in CV to improve the performance of feature extraction algorithms at the bottom level of the vision hierarchy [15]. These methods tolerate (in various degrees) the presence of data points that do not obey the assumed model. Such points are typically called "outliers". The definition of robustness in this context is often focused on the notion of the breakdown point: the smallest fraction of outliers in a data set that can cause an estimator to produce arbitrarily bad results. The breakdown point, as defined in statistics, is a worst-case measure. A zero breakdown point only means that there exists one potential configuration for which the estimator will fail.

ChatGPT is a completely new approach in the CV world [16]. It is a tool that can help CV engineers and practitioners to fulfill jobs efficiently. There are 3 main categories of CV applications for which ChatGPT is fairly reliable: commonplace code, dressed individual method calls, and clean concatenations of simple components. ChatGPT's responses to queries in any of these categories benefit from being relatively self-contained. A generative model that was trained on a large corpus, including text and code, is generally satisfactory at generating blocks of code that occur frequently and with little variation across the internet. When a code-based solution is essentially canonical (and likely omnipresent in the training data), ChatGPT's probabilistic predilections are able, with high probability, to generate the tried and true result. Examples of this include the fast Fourier transform (FFT), which is ubiquitous in signal processing and CV. ChatGPT is compatible with common machine learning and CV libraries, including PyTorch, TensorFlow, Scikit-learn, PIL, Skimage, and OpenCV. The chatbot is at its best when it can dress up methods from these libraries with the appropriate (boilerplate) preprocessing steps, such as input-output handling, converting a color image to grayscale, and reshaping arrays.

With new technology, the failure modes may be potentially powered. While applying the ChatGPT for multiple CV tasks, there seem to be a few recurring issues: long-tail scenarios, math manipulations, and expansive code blocks. There may happen a variety of tasks that are staples of certain subfields but are dwarfed by more common motifs in the sprawling corpora employed in training LLMs. ChatGPT has its fair share of trouble with these domains and can be quite sensitive to minutiae when prompted on niche subjects. One word can mean the difference between a desired result, and an idea getting lost in the recesses of ChatGPT's immense representational structure. An example of this is 3D computer vision, which seems to be a strong subfield of CcV that deals with spatial data.

The closer to our task - quality control, the more we apply a standard method with few details that depend on the specific situation. With a field as broad and complex as computer vision, the solution isn't always clear [17]. The many standard tasks in CV require special consideration: classification, detection, segmentation, pose estimation, enhancement and restoration, and action recognition. Although the state-of-the-art networks used exhibit common patterns, they still need their unique design twist.

Classification. Image classification networks start with an input of fixed size. The input image is provided by some channels, usually 3 for an RGB image. When you design the network, the resolution can technically be any size as long as it is large enough to support the amount of downsampling you perform by the network. For example, if you downsample 4 times within the network, then your input needs to at least be 16 x 16 pixels in size. As go deeper into the network the spatial resolution will decrease as we try to squeeze this information and get down to a 1-dimensional vector representation. To ensure that the network always can carry forward the information it extracts, increase the number of feature maps proportionally to the depth accommodating the reduction in spatial resolution. I.e., we are losing spatial information in the down-sampling process. To accommodate the loss, we expand our feature maps increasing the obtained semantic information. After a certain amount of downsampling has been selected, the feature maps are vectorized and fed into a series of fully connected layers. The last layer has as many outputs as there are classes in the dataset.

Object Detection. Object detectors come in 2 flavors: one-stage and two-stage [18]. Both of them start with “anchor boxes”; these are default bounding boxes. Our detector is going to predict the difference between those boxes and the ground-truth, rather than predicting the boxes directly. A two-stage detector naturally has two networks: a box proposal network and a classification network. The box proposal network proposes coordinates for bounding boxes where it thinks there is a high likelihood that objects are there; these are relative to the anchor boxes. The classification network then takes each of these bounding boxes and classifies the potential object that lies within it.

Segmentation. Segmentation is one of the most unique tasks in CV since the networks need to learn both low- and high-level information [19]. Low-level information to accurately segment each area and object in the image by the pixel, and high-level information to directly classify those pixels. This leads to networks being designed to combine the information from earlier layers and high-resolution (low-level spatial information) with deeper layers and low-resolution (high-level semantic information). First, we run our image through a standard classification network. Then we extract features from each stage of the network, transferring information to a higher level. Each information level is processed independently before combining in turn. As the information is combined, we upsample the feature maps to eventually get to the full image resolution.

Pose Estimation. These models need to accomplish 2 tasks [20]: (1) detect key points in an image for each body part and (2) learn how to properly connect these key points. It can be done in 3 stages:

- Extract features from the image using a standard classification network
- Given those features, train a sub-network to predict a set of 2D heatmaps. Each heatmap is associated with a particular key point and contains confidence values for each pixel in the image as to whether the key point is likely to exist there or not
- Again, with the given features, we train a sub-network to predict a set of 2D vector fields, where each vector field encodes the degree of association between the key points. Key points with the high association are then said to be connected.

Training the model in this way with the sub-networks jointly optimizes the detection of key points and connecting them.

5. Determination of quality criteria

As mentioned in the above point, most computer vision systems include the previously mentioned stages, but according to the specifics, different options are possible.

Definition of quality criteria.

A hybrid system offers an opportunity to integrate two or more knowledge representations of a certain area into one system. One of the main goals is to obtain additional knowledge that allows for increasing the efficiency of the global system. A concrete example of a hybrid system is the so-called NSHS, which is mainly based on a symbolic representation of an object obtained from a human expert in the form of rules and a CV system for obtaining numerical information.

Quality criteria for the assessment of pears were obtained by direct visual evaluation by an expert in fruit classification based on his own experience. In this case, the category is assigned depending on the value of the external attributes. There are four categories: category extra, category I, category II, and category III. The work evaluates only the extra category, according to which a pear can belong to one of two classes: good or bad quality. Fig. 3 shows an example of pears, which could be classified as good and poor quality, that is, the second pear has a clear visual defect. Additionally, Table 1 reveals a summary of the external attributes of the pear with the associated variable name, value, and type.

Getting an image.

For the pear category task, 148 images were obtained using a digital camera. For the complete set of images, the operation of rotation by 90 and 180° clockwise, doubling and reducing scaling, and adding noise were performed. At the end of this process, a set of 148 pears was divided into two categories: poor (74) and good (74) quality. Fig. 4 displays an example of different pears after changing the rotation and scale.

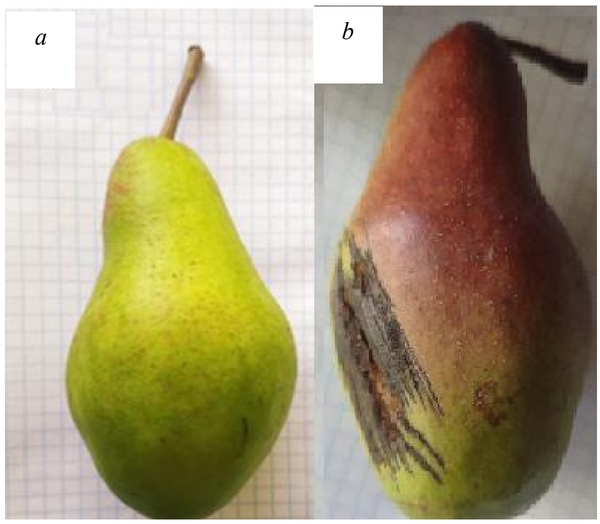

Fig. 1. Examples of a pear, a) good quality and b) poor quality with a clear visual defect

Table 1. Criteria defined in dimensionless units for establishing good and poor quality, obtained in consultation with relevant experts in the given field

Attribute	Acronym	Type	Value
An elongated defect	LD	Range	0–6
Spotted defect	SD	Range	0–2.5
Various defects	VD	Range	0–5
Stem	S	Binary	True/false
Red colour	RC	Range	0–255
Green	GC	Range	0–255
Blue	BC	Range	0–255

Pre-processing of images.

The stage consists of converting the image from the RGB color model to the YIQ color model (Luminance, In-phase, Quadrature). The main reason for this transformation is to facilitate the extraction of image features. The YIQ model was calculated to separate the color from the luminance component due to the ability of the human visual system to perceive changes in reflectance more than changes in hue or saturation. The main advantage of the model is that reflectance (Y) and color information (I and Q) can be processed separately. The reflection coefficient is proportional to the amount of light perceived by the human eye.

Pear feature extraction.

The characteristics of each image were obtained based on information defined by a human expert in the form of rules and by image processing in the form of numerical data. These two types of knowledge information were combined to obtain an overall view of the pear.

The number of rules defined by the experts was four, an example of one rule is the following: "If a pear has a suitable color, has a stem, has elongated defects not exceeding 2 cm, and has several defects not exceeding 1 cm², and has spotted defects, which do not exceed 1/4 cm², then the pear belongs to the category of extra with good quality.

At the end of this step, the rules were compiled using knowledge of the based artificial neural network to obtain the sample that can then be combined with the numerical results obtained from the CV system. The combination was done using a method called Neusim, which is based on the Fahlmann cascade correlation algorithm.

Classification of pears.

The result of the feature extraction stage is a combined representation of the symbolic and numerical representation. Further classification requires clarification of these data. This refinement is done by running the Neusim method again, but now not for knowledge fusion, but for using it as a classifier.

The main advantage of using the Neusim algorithm is that one can see the number of hidden units added during the learning process, this is quite useful for monitoring the complete incremental learning process. The result of this stage is a decision about the quality of the pear in one of two classes, bad or good.

Fig. 2. Good quality pear. a) Original image, b) Rotated 180° clockwise, and c) Halved original size

Experiments and results.

From the total set of 148 images, 74 were used at the training stage and 74 at the recognition stage. Three different approaches were chosen for the experiments: (a) a connectionist approach, which uses only data obtained from a computer vision system, (b) a symbolic approach, which uses only data obtained from collected rules, and (c) NSHS, which is a combination of connectionist and symbolic approach.

Three scenarios were defined for tests using the connectionist approach: (a) numerical data obtained from a total of 148 images (100%), (b) data obtained from only 111 images (75%), and (c) only data from 74 images (50%). Three rules were applied to obtain results relevant to the test case using a symbolic approach. The first rule, called 7, includes the following seven attributes: LD, SD, VD, S, RC, GC, and BC. The second rule, called R5, considers the following five attributes: RC, GC, BC, S, and LD. Finally, the third rule named R4 includes the following four attributes: LD, SD, VD, and S. For the case of the NSHS approach, a combination of connectionist and symbolic approaches. The three rules R7, R5, and R4 were combined with 100, 75, and 50% of the total examples. The general results obtained are shown in Table 2.

Table 2. The achieved results

Approach	Compiled rules	% of examples used	percentage of correct decisions (%)
Connectionist	–	100	95.14
	–	75	91.21
	–	50	90.54
Symbolic	R7	–	93
	R5	–	90.12
	R4	–	14.19
NSHS	R7	100	96.62
	R7	75	95.27
	R7	50	90.54
	R5	100	95.27
	R5	75	95.94
	R5	50	96.62
	R4	100	91.22
	R4	75	93.24
	R4	50	94.59

One of the typical problems causing failures in CV systems is the lack of a complete description of the object. This can be observed by examining the results based on the symbolic and connectionist approaches. This shortcoming can be eliminated by using a method to supplement the information with data determined by the expert's knowledge. Systems that allow these types of combinations are called NSHS, as can be seen from the results shown in Table 2; these systems are effective in supplementing the necessary knowledge for automatic object inspection. For example, in a purely symbolic approach, the R4 rule was not sufficient for correct classification, but when it is integrated with a group of numerical examples (100, 75, 50%), a significant improvement is obtained, since the knowledge that does not contain the rule is supplemented by a numerical example base.

Here, satisfactory results are achieved in almost all cases. This indicates that a considered method can be implemented to define the quality of a wider list of products.

6. Conclusions

1. A computer vision-based quality inspection system was studied to create a sustainable evaluation environment in the fruit quality inspection process. To access the quality of fruits, an image-learning model adopting an artificial neural network was developed.

2. While performing the task, during the verification of the results practicing symbolic and connectionist approaches, it is possible to observe failures in computer vision systems due to the lack of a complete description of the object. This shortcoming can be overcome by supplying the obtained information with data determined by expert knowledge, proving systems that allow these types of combinations, or/and the NSHS approach itself.

3. Due to the proposed different approaches in image processing, a high level of truthfulness is achieved, which is expressed within the range from 90 to 96%. The obtained results can automate the process of visual inspection with the prospect of increasing the speed and quality of product sales for the consumer.

7. Acknowledgment

The authors express their gratitude to the staff of the Information-measuring technologies department of the Lviv Polytechnic National University, Ukraine, for the help and assistance provided in the preparation of this article.

8. Conflict of interest

There were no financial, organizational, or other possible conflicts during the performance of the work.

References

[1] J. García-Alcaraz, A. Maldonado-Macías, G. Cortes-Robles. Lean Manufacturing in the Developing World,

117(8), 827–891. 2014, Available: https://link.springer.com/book/10.1007/978-3-319-04951-9

[2] W. Sullivan, T. McDonald, & Van Aken, E. (2002). Equipment replacement decisions and lean manufacturing.Robotics and Computer Integrated Manufacturing, 18(3–4), 255–265. Available: https://www.sciencedirect.com/science/article/abs/pii/S0736584502000169

[3] G. Kreiman. Biological and Computer Vision 52-53 [2021]. Available: https://www.cambridge.org/ core/books/biological-and-computer-vision/BB7E68A69AFE7A322F68F3C4A297F3CF

[4] R. Szeliski. Computer Vision: Algorithms and Applications. 42-43. 2022. Available: https://szeliski.org/Book/

[5] J. Watt, R. Borhani, A. Katsaggelos. Machine Learning Refined: Foundations, Algorithms, and Applications 45-46 [2020]. Available: https://people.engr.tamu.edu/guni/csce421/files/Machine_Learning_Refined.pdf

[6] A. Koul, S. Ganju, M. Kasam. Practical Deep Learning for Cloud, Mobile, and Edge: Real-World AI & Computer-Vision Projects Using Python, Keras & TensorFlow. 25-26. 2019. Available: https://www.oreilly.com/library/view/practical-deep-learning/9781492034858/

[7] J. Kelleher. Deep Learning 12-13 [2019]. Available: https://direct.mit.edu/books/book/4556/Deep-Learning

[8] T. Amaratunga. Deep Learning on Windows: Building Deep Learning Computer Vision Systems on Microsoft Windows 88-89 [2021]. Available: https://link.springer.com/book/10.1007/978-1-4842-6431-7

[9] M. Favorskaya, L. Jain. Computer Vision in Control Systems. 55-56. 2015. Available: https://link.springer.com /book/10.1007/978-3-319-10653-3

[10] K. Mehrotra, Ch. Mohan, H. Huang. Anomaly Detection Principles and Algorithms. 165-166. 2017. Available: https://www.amazon.com/Detection-Principles-Algorithms-Terrorism-Computation/dp/3319675249

[11] S. Prince. Computer Vision: Models, Learning, and Inference. 222-223. 2022. Available: http://www. computervisionmodels.com

[12] G. Garrido, P. Joshi. OpenCV 3.x with Python By Example, 33-34 [2018]. Available: https://www.packtpub.com/product/opencv-3x-with-python-by-example-second-edition/9781788396905

[13] N. Madanchi. Model-Based Approach for Energy and Resource Efficient Machining Systems. 111-112. 2022. Available: https://link.springer.com/book/10.1007/978-3-030-87540-4

[14] M. Rafiei. Self-Supervised Learning A-Z: Theory & Hands-On Python. 24-25. 2022. Available: https://www.udemy.com/course/self-supervised-learning/

[15] N. Sebe, M. Lew. Robust Computer Vision. 54-55. 2022. Available: https://link.springer.com/book/10.1007/978-94-017-0295-9

[16] J. Marks. Tunnel vision in computer vision: can ChatGPT see?. 2022. Internet Journal. Available: https://medium.com/voxel51/tunnel-vision-in-computer-vision-can-chatgpt-see-e6ef037c535

[17] A. Rosebrock. Deep Learning for Computer Vision with Python. 55-56. 2017. Available: https://bayanbox.ir/view/5130918188419813120/Adrian-Rosebrock-Deep-Learning-for.pdf

[18] G. Blokdyk. Object detection Complete Self-Assessment Guide. 42-43. 2021. Available: https://www.amazon.com/dp/0655324380?tag=uuid10-20

[19] R. Shanmugamani. Deep Learning for Computer Vision. 68-69. 2021. Available: https://www.oreilly.com/ library/view/deep-learning-for/9781788295628/edf4fbcc-cca0-4aaa-b9f4-d7c2292c520d.x.html

[20] J. Howse, P. Joshi, M. Beyeler. OpenCV: Computer Vision Projects with Python. 122-123. 2020. Available: https://www.oreilly.com/library/view/opencv-computer-vision/9781787125490/ch16s04.html