arXiv:2011.05155v2 [eess.IV] 27 Dec 2021

Machine learning based luminance analysis of a µLED array

Steven Becker — steven.becker@tu-dortmund.de

Experimentelle Physik 2, TU Dortmund University
Otto-Hahn-Straße 4a, 44227 Dortmund, Germany

December 30, 2021

In the past years, the development of µLED arrays gained momentum since they combine the advantages of LEDs, such as high brightness and longevity, with the high resolution of a micro-scaled structure. For their development, spatially resolved measurement of luminance and color of single µLEDs and the entire light-emitting surface are usually analyzed as they quantify the visual perception. However, studying individual µLEDs is time-consuming to measure and evaluate, while examining the entire light-emitting area suffers from interference from non-functioning µLEDs. This paper presents a method to perform both analyzes with a single measurement employing unsupervised machine learning. The results suggest that a precise reconstruction of the µLEDs and a more accurate characterization µLED arrays is achieved.

Introduction

Over the last decades, light-emitting diodes (LEDs) were established as a key light source in the consumer market, such as for general lighting or automotive. The accelerators for the far-reaching influence of LEDs are their durability, even under harsh conditions such as extreme temperatures, and their high-efficiency and luminance. However, for applications requiring high pixelation like displays, the high-performance LEDs are mostly too big because research has focused on the optimization of $1\,\mathrm{mm}^2$ large high-performance LEDs [1], [2].

Shrinking those LEDs into the micro-scale - edge length below 100 µm- allows to overcome this issue and should theoretically increase their efficiency. Already in 2000, Jin et al. published the first realizations of micro light-emitting diodes µLED [3], and since then, numerous publications regarding this topic were submitted.

One advantage of µLEDs is the possibility of combining them to an array-structure, which can then be linked with an underlying electrical control unit. This two-dimensional stringing-together of µLEDs results in a µLED array. Such structures should have a higher illuminance and homogeneity than a single µLED, as well as provide a high brightness, contrast, resolution, and durability [4], [5]. In combination, they may become superior against already established pixelated light sources such as organic LEDs (OLEDs) or liquid crystal (LC) based ones [2], [6]. Furthermore, LED arrays gained much momentum in the industry as well [6]. However, there is no published data that the established manufacturing techniques, mass transfer-based or the monolithic-based fabrication, achieved a yield of close to 100% [7]. Consequently, most of the produced µLED arrays will contain non-emitting and therefore defect µLEDs.

Classical approaches to characterize the luminance of the light-emitting surface (LES), such as averaging over the entire area, do not distinguish between functional and defect, resulting in an underestimation of the actual µLED array behavior. If a significant fraction of nonfunctional µLEDs occurs, this becomes a problem in the development process. For instance, the "noise" created by the defect µLEDs could prevent an evaluation of a design change. Each µLED (pixel) should be classified, and defect ones may not be considered for the final analysis to overcome this blurring effect.

In this paper, a machine learning-based algorithm is applied to overcome this issue. Therefore, individual µLEDs are located in a luminance image of a µLED array and then classified as functional or defect with an unsupervised learner (KMeans). As a result, the behavior of the µLED array is studied without the blurring effect of the non-emitting µLEDs. The implemented analysis is evaluated by three performance measures including the reconstructed µLED size, the confusion matrix of the underlying classifier, and the noise reduction performance. In order to achieve this, a luminance camera with a resolution of 2448×2050 pixels and a white light-emitting µLED array with a total of 60×60 µLEDs are used. Each µLED of the light source has a size of 40 µm $\times$ 40 µm and can be addressed separately. A lens with a magnification factor of two is mounted on the camera. Due to the luminance camera pixel's size of 3.45 µm, a single µLED is represented by approximately 23 camera pixels.

Results

The developed analysis is tested with four luminance images, which are shown in Figure 1. The first image represents the *Reference state*, where all µLEDs of the array are turned on. The non-uniform luminance distribution originates from the underlying electronics and will not be discussed further. On the other images (*Pixel map* one to three), some µLEDs are turned-off on purpose to simulate a nonfunctional state. These pseudo-defect µLEDs were selected for each *Pixel map* randomly and represent a yield of 80%, hence 20% of the µLEDs are turned-off. Furthermore, the four luminance images are the direct outputs of the luminance camera, which explains the black bars framing the array

structure. In addition, the sample was rotated to increase the difficulty and to stress the robustness of the method.

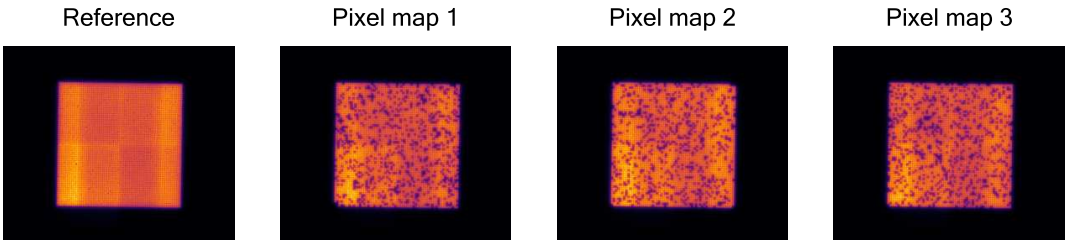

Figure 1: Luminance images of the used *Pixel maps* to benchmark the analysis pipeline.

The *Pixel maps* shown in Figure 1 are the inputs of the underlying analysis pipeline, which is illustrated in Figure 2. The first step of the pipeline performs a *projective transformation* to compensate for a tilted and rotated sample. Then, the *grid reconstruction* scopes to reassemble the array structure, so that individual µLEDs can be localized. This result is exploited by the *Pixel analysis* that studies the luminance and CIE color coordinates of each reconstructed pixel. The acquired data set is then guided into the machine learning part of the pipeline, which utilizes a *principal component analysis* (PCA) and an unsupervised learner (KMeans) to classify functional and nonfunctional µLEDs. Finally, the *Array level analysis* evaluates the entire light-emitting surface (LES), while considering the information about the functionality of the individual µLEDs. Note, each step of the pipeline will be discussed in more detail in the subsequent section. In total, three performance measures can be extracted from the analysis pipeline (see Fig. 2): Firstly, the reconstructed pixel size, secondly the confusion matrix of the pixel classification, and finally the mean luminance of the LES.

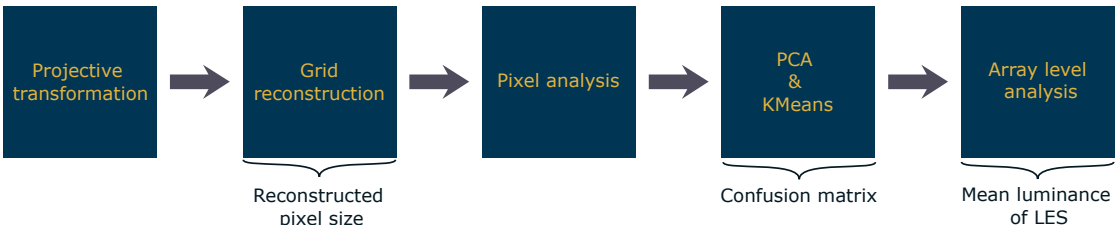

Figure 2: Schematic illustration of the analysis pipeline.

Since the luminance image provides spatial information of the µLED array, the pixel grid framing the individual µLEDs can be extracted. Figure 3 shows the result of the underlying grid reconstruction algorithm for *Pixel map 1*. In addition, Figure 3 confirms that the applied projective transformation can correct the initial rotation of the sample. The method is even capable of reconstructing the grid at locations where several pixels

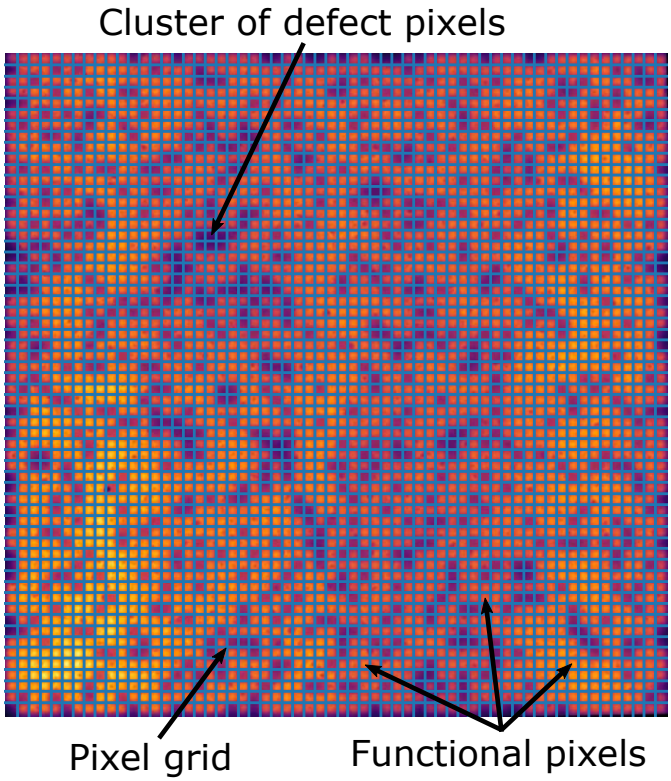

Figure 3: Reconstructed pixel grid (blue lines) of the *Pixel map 1* (see Figure 1).

are "defect". The quality of the extracted grid can be measured by its pixel size since the pixel size of the grid ideally corresponds to the number of camera pixels representing a µLED. Equation (1) shows the mean grid's pixel size $\overline{d}$ for all three *Pixel maps* (compare Figure 1):

$$\overline{d}_{\text{pixel},1} = \overline{d}_{\text{pixel},2} = \overline{d}_{\text{pixel},3} = (23 \times 23)\ \text{px}^2. \tag{1}$$

The standard deviation of the mean value is intentionally neglected because the uncertainty is less than 0.5 px. As a result, it is concluded that the pixel grid reconstruction is capable of consistently reconstructing the proposed pixel size. Note that pixels at the edges of the LES are currently not considered to diminish the influence of boundary effects.

Next, the reconstructed grid organizes the input luminance image (compare Figure 1) into individual pixel areas containing information about a certain µLED. Hence, from a single luminance image, information about hundreds of µLEDs can be extracted and used for statistical analysis. The maximum, mean, minimum, and standard deviation of the luminance as well as the mean CIE x and CIE y color coordinates are calculated for each pixel including the "defect" ones. Consequently, this procedure leads to a statistical data set that not only represents a base for a profound analysis of the µLEDs behavior but also serves here as training data for the unsupervised learner. In partic-

ular, the classification of defect pixels is of interest as elaborated in the introduction. The achieved performance of the used *k-Means-algorithm* is illustrated in the confusion matrices shown in Figure 4. The confusion matrices reveal for each *Pixel map* (compare Figure 1) the corresponding prediction accuracy for the classification *functional / defect*. Note that the general procedure of the classification step is explained in detail in the subsequent section. According to the matrices, the classifier predicts the pixel's

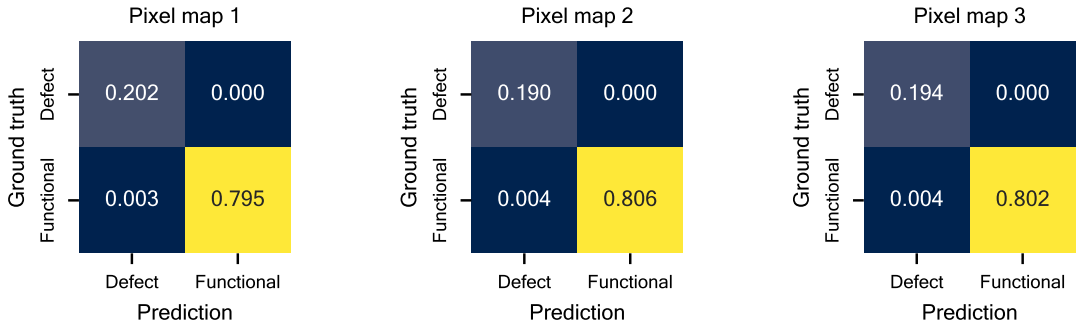

Figure 4: Confusion matrices for the pixel classification of the three *Pixel maps*. The pipeline achieves an accuracy of $\approx 99.5\%$ for each *Pixel map*, as indicated through the diagonal elements. Further, the lower left and upper right matrix elements reveal a false negative rate of under 0.5% and a false positive rate of 0%, respectively.

status with a percentage of $\approx 99.5\%$ correct. In the other case, the lower left matrix elements indicate that functional pixels are falsely classified as defective in less than 0.5%. Moreover, the obtained true positive and negative rates of about 80% and 20%, respectively, are consistent with the originally selected yield of 80%. In general, from the confusion matrix further evaluation quantities can be extracted including, but not limited to, the precision score P, recall score R, and the $F1$-score. The precision score measures the ability of a classifier to prevent the positive labeling of something that should be negative. In the case of this classifier, it quantifies the risk of predicting a defect pixel as functional. The recall score represents the capability of a classifier to identify all positive elements, here the ability to notice functional pixels. The $F1$-score considers both precision and recall score to give a balanced quantity [8]. Note, for all scores 0 is the worst and 1 is the best outcome. The scores can be calculated as follows:

$$\begin{aligned} TP &= \text{True positiv}, \qquad FP = \text{False positiv}, \qquad FN = \text{False negative} \\ P &= \frac{TP}{TP+FP}, \qquad R = \frac{TP}{TP+FN}, \qquad F1 = 2\frac{P \cdot R}{P+R}. \end{aligned} \tag{2}$$

The calculation of those measures confirms the performance of the *k-Means*-based clas-

sifer:

$$P_1 = P_2 = P_3 \approx 0.995 \approx 1 \tag{3}$$
$$R_1 = R_2 = R_3 = 1 \tag{4}$$
$$F1_1 = F1_2 = F1_3 = 1. \tag{5}$$

Finally, the results can be exploited to extract information about the LES even with nonfunctional pixels. As elaborated in the introduction, averaging over the LES, which includes nonfunctional pixels, lead to an underestimation of the actual performance of the LES. In order to quantify the improvement of the presented analysis, the mean luminance of the *Reference state* (left in Figure 1) is determined: $\overline{L}_{\text{ref}} = (286.4 \pm 1.1) \times 10^4 \, \frac{\text{cd}}{\text{m}^2}$. The mean luminance analysis for all *Pixel maps* yields the following:

$$\begin{aligned}
L_{1,\text{raw}} &= (256.7 \pm 0.8) \times 10^4 \, \frac{\text{cd}}{\text{m}^2}, \quad \overline{L}_{1,\text{dn}} = (275.2 \pm 0.6) \times 10^4 \, \frac{\text{cd}}{\text{m}^2} \\
\overline{L}_{2,\text{raw}} &= (257.0 \pm 0.8) \times 10^4 \, \frac{\text{cd}}{\text{m}^2}, \quad \overline{L}_{2,\text{dn}} = (273.9 \pm 0.5) \times 10^4 \, \frac{\text{cd}}{\text{m}^2} \\
\overline{L}_{3,\text{raw}} &= (258.2 \pm 0.8) \times 10^4 \, \frac{\text{cd}}{\text{m}^2}, \quad \overline{L}_{3,\text{dn}} = (275.5 \pm 0.6) \times 10^4 \, \frac{\text{cd}}{\text{m}^2}.
\end{aligned} \tag{6}$$

The index *raw* indicates the mean luminance, including the defect pixels, whereas the index *dn* marks the mean luminance only for functional µLEDs. Consequently, the proposed analysis pipeline is capable of reconstructing a more representative value for the mean luminance compared to the classical approach.

Conclusion

The presented analysis technique is capable of reconstructing the pixel grid with high precision, even when clusters of µLEDs are not working. Further, the machine learning-based classification of the current pixel status also shows a high overlap with the simulated yield. In comparison, to a classical classification approach using, for instance, an arbitrary luminance threshold, no intense parameter tuning is required. Finally, exploiting the knowledge about the µLED status enhances the analysis of the entire LES and allows to infer the actual behavior of the LES more preciously as a classical approach. However, since the current pipeline uses an unsupervised learner (KMeans), it could behave differently on different µLED arrays. Swapping to a supervised learner such as a RandomForest could reinforce the robustness of the analysis, however, requires the presence of a labeled data set, which is time-intense to accumulate. Moreover, being able to classify thousands of µLEDs within a single measurement also offers the possibility to study the statistical distribution of each quantity for both functional and nonfunctional µLEDs.

Methods

Luminance measurement

The used setup, shown in Figure 5, employs the luminance camera LMK5-5 of the manufacturer TechnoTeam. The camera offers a resolution of 2448×2050 pixels and is equipped with a filter wheel to consider the $V(\lambda)$ curve and color-weight functions for the measurements of luminance and color, respectively. Besides, the luminance camera uses a lens with a magnification factor of two. On the sample side, a white light-emitting

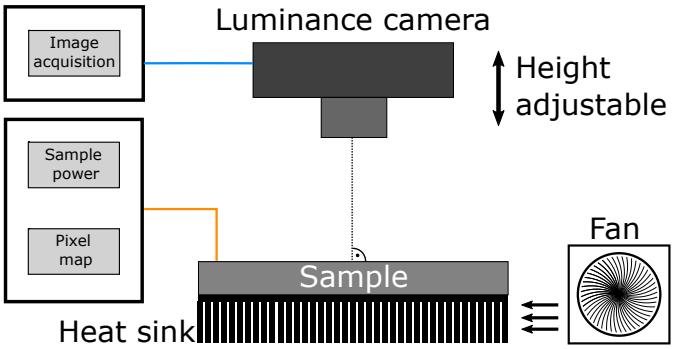

Figure 5: Schematic illustration of the used luminance setup. The height-adjustable luminance camera is aligned perpendicular to the LES. The sample is mounted on a heat sink, which is cooled actively with a fan.

µLED-array with a total of 60×60 µLEDs, where each has size of $40\,\mu\text{m} \times 40\,\mu\text{m}$, is used. Due to the luminance camera pixel's size of $3.45\,\mu\text{m}$, a single µLED is represented by approximately 23×23 camera pixels. Accordingly, the setup is theoretically capable of resolving smaller µLEDs. However, the significance of the pixel-level analysis diminishes with fewer camera pixels per µLED. For instance, local anomalies on a µLED would be averaged out with fewer camera pixels, yielding a less accurate statistical representation of the µLED array's behavior. Furthermore, the grid reconstruction reaches a resolution limit for fewer camera pixels. Tests with a narrowing lens (size was halved) indicate that the lower threshold for the algorithm lays at a single µLED representation of at least with 10×10 camera pixels, beyond this point the accuracy of the grid can diminish.

Image correction

The projection transformation is a technique to project an input image into an equivalent image, keeping its properties using a linear transformation. A deeper insight in the mathematical description of the method is given in reference [9]. Figure 6 shows how this transformation removes projective distortions (tilting, rotation) from an image. Note that tilting can not only occur from a miss-aligned camera but also from the µLED array itself. The perspective transformation is performed using CV2.GETPERSPECTIVETRANSFORM

and CV2.WARPPERSPECTIVE, which are implemented in the PYTHON package CV2 (based on OPENCV) [10], [11]. The edges of the LES are located in the luminance image by thresholding the input image with CV2.THRESHOLD and then routing the output into the contour locator CV2.FINDCONTOURS.

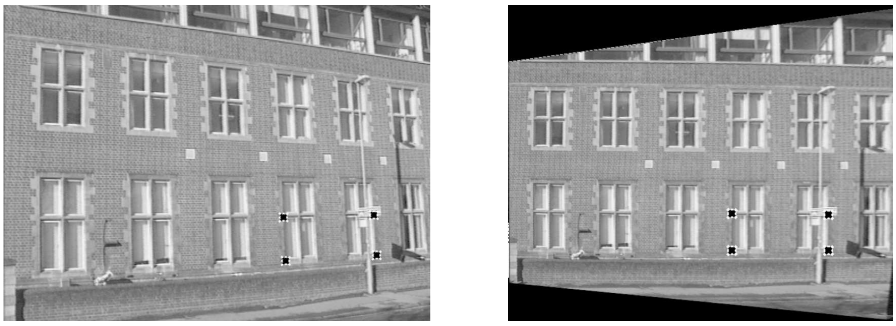

Figure 6: Illustration on how a projection transformation can remove tilting and rotation [9].

Grid reconstruction

After ensuring an adequate alignment, the pixel grid is reconstructed by projecting the luminance image on the x- and y-axis, which can be formally written as:

$$\begin{aligned} L(x,y) &\in \mathbb{R}^{N\times M}, \quad \text{Luminance image} \\ \Sigma_{\mathrm{x}} L(x_i) &= \sum_{j=1}^{M} L(x_i, y_j)\ \forall i \in [1,N]\,, \quad \mathrm{x}-\text{projection} \\ \Sigma_{\mathrm{y}} L(y_j) &= \sum_{i=1}^{N} L(x_i, y_j)\ \forall j \in [1,M]\,, \quad \mathrm{y}-\text{projection}. \end{aligned} \tag{7}$$

As indicated by Figure 7 the values for $\Sigma_{\mathrm{x}} L(x_i)$ and $\Sigma_{\mathrm{y}} L(y_j)$ are significantly smaller at the edge of two µLEDs, leading to minima. Through the minima locations in the projections, it is possible to detect the pixel edges and, therefore, localize the pixel grid. Moreover, since the minima are independent of the µLEDs itself, the presented method can reconstruct the pixel grid even at positions where multiple pixels are defective. Figure 3 shows the reconstructed grid for the first *Pixel map*. Remarkable is that the pipeline reconstructs the pixel position even for a cluster of nonfunctional pixels correctly.

Pixel classification

With the pixel location, a pixel-level analysis can be performed, whereby the exact type of analysis can be adapted for different use cases. In the case of the subsequent pixel

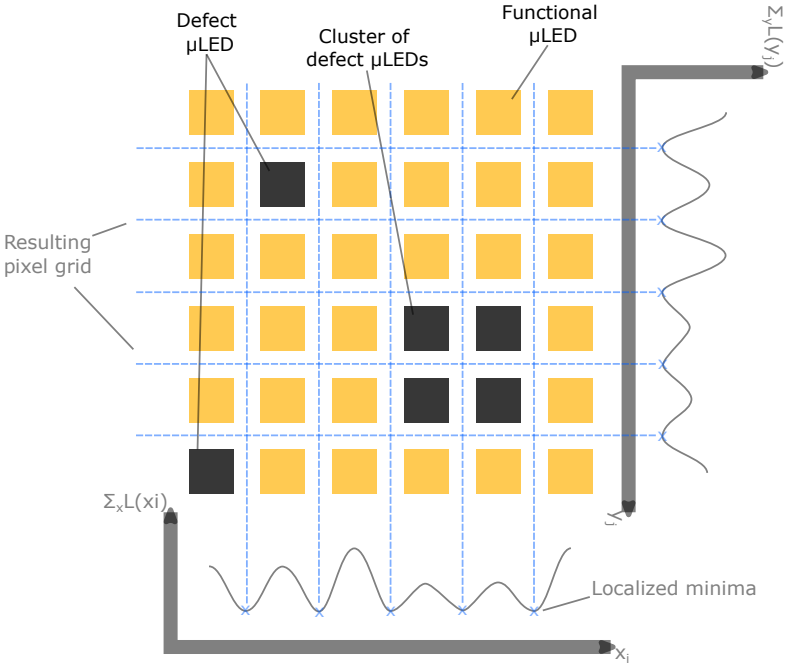

Figure 7: Illustration on how the projection of the luminance on to the x- and y-axis enables the reconstruction of the pixel grid.

classification, the mean, maximum, minimum, and standard deviation of the pixel's luminance are extracted. Additionally, the mean color coordinates CIE x, and CIE y are also calculated. In total, six parameters describe a single µLED (see Figure 8), creating a six-dimensional parameter space. This parameter space is exploited for machine learning. Although this would be ideal for training a supervised learner such as a RANDOM

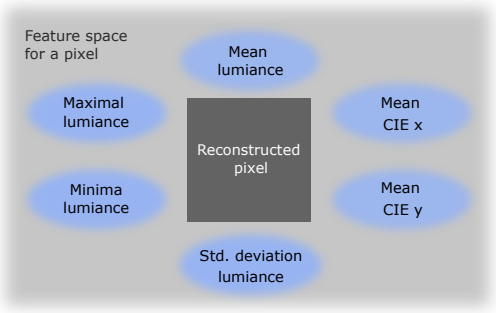

Figure 8: Visualization of the feature set for each reconstructed pixel leading to a six-dimensional parameter space.

FOREST, the lack of labeled data motivated the use of an unsupervised learner. From the variety of promising methods, such as *Gaussian Mixture Models*, a simple *k-Means-algorithm* was chosen because it offers for this classification task a precision of over 99%.

In addition, its simplicity allows people with less background in machine learning to quickly implement the presented analysis pipeline.

After extracting the information shown in Figure 8 for each pixel, the data is standardized with SKLEARN.PREPROCESSING.SCALE. Subsequently, a *principal component analysis* (PCA) aims to increase the information density of the feature set by reducing the number of the feature space to two. A PCA tries to find new axes in the input parameter space, which maximize the variance of the data. These axes correspond to the eigenvectors with the largest eigenvalues extracted from the samples input covariance matrix. After determination of the eigenvectors, the input data is projected onto these new axes [12]. The reference [13] provides a mathematical description. On the software side, the implementation SKLEARN.DECOMPOSITION.PCA of the PYTHON package SCIKIT-LEARN is applied [14].

The outcome of the PCA is then guided into the actual classifier, which is a *k-Means-algorithm*. This algorithm tries to classify the input data into k clusters by minimizing the squared distance of all data point to the cluster centers. In the context of this paper, k equals two. A more detailed description can be found in reference [13]. For the analysis, the implementation SKLEARN.CLUSTER.KMEANS of SCIKIT-LEARN with a random seed of eight and $n_{\text{init}} = 100$ (number of repetitions) is utilized.

Acknowledgements

The author thanks OSRAM Opto Semiconductors GmbH for providing the light source. In addition, the author thanks Prof. Dr. Manfred Bayer and the chair Experimentelle Physik 2 of TU Dortmund University for supporting this work.

Disclosures

The authors declare no conflicts of interest.

Author contributions statement

S.B. conceived the experiments, developed and implemented the analysis, and reviewed the manuscript.

References

[1] H. S. Wasisto, J. D. Prades, J. Gülink, and A. Waag, "Beyond solid-state lighting: Miniaturization, hybrid integration, and applications of gan nano- and micro-leds," *Applied Physics Reviews*, vol. 6, no. 4, p. 041 315, 2019. DOI: 10.1063/1.5096322.

[2] J. Day, J. Li, D. Y. C. Lie, *et al.*, "Full-scale self-emissive blue and green microdisplays based on GaN micro-LED arrays," in *Quantum Sensing and Nanophotonic Devices IX*, M. Razeghi, E. Tournie, and G. J. Brown, Eds., International Society for Optics and Photonics, vol. 8268, SPIE, 2012, pp. 428–435. DOI: 10.1117/12.914061. [Online]. Available: https://doi.org/10.1117/12.914061.

[3] S. X. Jin, J. Li, J. Z. Li, *et al.*, "GaN microdisk light emitting diodes," *Applied Physics Letters*, vol. 76, no. 5, pp. 631–633, 2000. DOI: 10.1063/1.125841.

[4] J. Day, J. Li, D. Y. C. Lie, *et al.*, "III-nitride full-scale high-resolution microdisplays," *Applied Physics Letters*, vol. 99, no. 3, p. 031 116, Jul. 18, 2011, ISSN: 0003-6951, 1077-3118. DOI: 10.1063/1.3615679. [Online]. Available: http://aip.scitation.org/doi/10.1063/1.3615679 (visited on 05/19/2020).

[5] C. Tian, S.-x. Guo, J.-q. Liang, *et al.*, "Effects of unit size on current density and illuminance of micro-LED-array," *Optoelectronics Letters*, vol. 13, no. 2, pp. 84–89, Mar. 2017, ISSN: 1673-1905, 1993-5013. DOI: 10.1007/s11801-017-7002-0. (visited on 05/19/2020).

[6] M. Y. Soh, W. X. Ng, T. H. Teo, *et al.*, "Design and characterization of micro-led matrix display with heterogeneous integration of gan and bcd technologies," *IEEE Transactions on Electron Devices*, vol. 66, no. 10, pp. 4221–4227, 2019.

[7] M. Wong, S. Nakamura, and S. DenBaars, "Review—progress in high performance iii-nitride micro-light-emitting diodes," *ECS Journal of Solid State Science and Technology*, vol. 9, p. 015 012, Jan. 2020. DOI: 10.1149/2.0302001JSS.

[8] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, 2020, ISSN: 2634-1964. DOI: 10.1016/j.aci.2018.08.003. [Online]. Available: https://doi.org/10.1016/j.aci.2018.08.003 (visited on 01/06/2021).

[9] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2. ed., 13.pr. Cambridge: Cambridge Univ. Press, 2015, 655 pp., OCLC: 947203565, ISBN: 978-0-521-54051-3.

[10] O.-P. Heinisuo, *Opencv-python*, PyPi (2020) [retrieved 12 August 2020], https://pypi.org/project/opencv-python/.

[11] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000. [Online]. Available: https://www.drdobbs.com/open-source/the-opencv-library/184404319 (visited on 08/07/2020).

[12] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, Feb. 1999, ISSN: 0899-7667, 1530-888X. DOI: 10.1162/089976699300016728. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/089976699300016728 (visited on 07/10/2020).

[13] C. M. Bishop, *Pattern recognition and machine learning*, 1st ed., ser. Information science and statistics. New York, NY: Springer, 2006, 738 pp., ISBN: 978-0-387-31073-2. [Online]. Available: https://bit.ly/3htZINY (visited on 12/07/2020).

[14] L. Buitinck, G. Louppe, M. Blondel, *et al.*, *Api design for machine learning software: Experiences from the scikit-learn project*, 2013. arXiv: 1309.0238 [cs.LG].