

A Neural Network for Interpolating Light-Sources

Simon Colreavy-Donnelly*, Stefan Kuhn[†], Fabio Caraffini[‡], Stuart O'Connor[§],
Zacharias A. Anastassi[¶] and Simon. Coupland

School of Computer Science and Informatics, De Montfort University, Leicester, UK

*simon.colreavy-donnelly@dmu.ac.uk (corresponding author) 0000-0002-1795-6995, [†]0000-0002-5990-4157,
[‡]0000-0001-9199-7368, [§]0000-0001-9029-269X, [¶]0000-0001-9190-2816

Abstract—This study combines two novel deterministic methods with a Convolutional Neural Network to develop a machine learning method that is aware of directionality of light in images. The first method detects shadows in terrestrial images by using a sliding-window algorithm that extracts specific hue and value features in an image. The second method interpolates light-sources by utilising a line-algorithm, which detects the direction of light sources in the image. Both of these methods are single-image solutions and employ deterministic methods to calculate the values from the image alone, without the need for illumination-models. They extract real-time geometry from the light source in an image, rather than mapping an illumination-model onto the image, which are the only models used today. Finally, those outputs are used to train a Convolutional Neural Network. This displays greater accuracy than previous methods for shadow detection and can predict light source-direction and thus orientation accurately, which is a considerable innovation for an unsupervised CNN. It is significantly faster than the deterministic methods. We also present a reference dataset for the problem of shadow and light direction detection.

Index Terms—shadow detection, light source detection, single-image solution, deep learning, unsupervised learning

I. INTRODUCTION

This article presents an improved shadow detection algorithm employing a Convolutional Neural Network (CNN) architecture which can distinguish light-source direction. This CNN, called the Full CNN hereafter, can accurately determine the difference between light falling on one side of an object, from another. It does this without manually annotated training data and faster than deterministic algorithms. This CNN, being aware of an abstract concept, is the main innovation of the paper. It generates real-world geometry from images, using a shadow detection technique. The CNN is unsupervised, as it utilises the output of two improved deterministic shadow and light direction detection algorithms that we developed for this purpose. These algorithms are used to generate the training data, thus removing the need for manual annotation.

This creates several very important potential uses for the method, as it is generating real-world directional information about light-sources in the scene, rather than projecting information onto the image as a means to determine light-direction, as is the case with present illumination-models. Therefore our method has value in creating navigation systems for robots, autonomous vehicles, augmented reality, and computer graphics-pipeline applications. Ultimately, understanding concepts like shadows and direction in an image, as opposed to only comparing the 2D values like colour variations, can

give a better understanding of the situation. A potential end goal would be the reconstruction of 3D geometries from 2D images.

Shadow detection poses challenges for image processing and has grown as a significant topic of investigation for a number of areas, including computer vision and computer graphics [1], [2], [3], [4], [5], [6]. When dealing with terrestrial images, shadows are a naturally occurring phenomenon that can become problematic for analysis even when focusing towards single image estimation [7]. This can lead to certain issues for extraction, segmentation and recognition, due to the changes in visual properties that shadows cause. It is therefore not surprising that accurate shadow detection is a difficult proposition. Various methods have been presented over the years, with different advantages and disadvantage [8], and their selection depends on the specifics of the application in question [9], [10]. CNNs in particular, have previously shown success when being employed in image classification and segmentation tasks, e.g. for medical purposes [11], [12], [13]. They naturally lend their application for shadow detection without requiring specific information regarding light sources or scene dynamics[14], [15] and thus show potential for further exploration to improve accuracy.

The light-source direction concept is quite abstract; however, in many ways, it is intrinsically linked with shadow detection. Existing machine learning methodologies in this area rely on pre-existing illumination models, or a known Azimuth (solar-direction, or direction of the Sun) [4]. When directionality is considered and utilised as a product of a CNN using a shadow detection technique with minimal training supervision, there are several potential applications due to the generated information. For augmented reality, whereby virtual objects are overlaid onto a real scene, consistency of light-source direction and shadow representation is an important consideration for achieving a realistic augmented final image [16], [17]. Computer vision applied towards the navigation of robots and autonomous vehicles highlight a particular avenue, with developments in the form of CNN's mapping raw pixels from a camera directly to steering commands showing promising results [18], [19]. It is noted that robustness and accuracy are significantly enhanced when identifying optimal parameters for properties including illumination conditions. This serves to highlight the potential of a CNN able to extract geometry from the light source within an image without the need for illumination-models.

It is worth noting that there are also a number of non-CNN-based light direction methodologies [20], [21], [22], [23], [24]. However, these have become exceedingly rare in comparison to CNNs and are similar to illumination-models found in shader-based computer graphics applications, such as standardised normal maps and Blinn-Phong-based illumination-models [25]. The Blinn-Phong illumination model (see [26] for implementation details, e.g. source code), is used in graphics pipeline applications, relying on the GPU and has been used since the 1990s for calculating shader programmes, as part of fragment and vertex-based pixel-shading in the now-ubiquitous *Nvidia* graphics card, with *OpenGL*. There is some providence in the fact that this technology is now also used extensively for machine learning problems[27].

II. METHODS

A convolutional neural network architecture works by analysing regions of the image, applying filters to those regions. The results are then aggregated using sub-sampling and pooling [28]. For training, which means learning the filters and weights, correct results, provided by experts, for some images are passed along with the original RGB image to the CNN. In the case of Neural Networks used in image classification for medical purposes, such as the U-Net [29], the result is a mask, labelling regions of interest. This methodology is used in this paper, except the masks are provided by an automated process. This process has two elements, or algorithms: 1) the sliding-window algorithm and 2) the line-direction algorithm. This training process is shown in the upper part of Fig. (1). The application of the model is only using the trained CNN, called the Full CNN hereafter, as shown in the lower part of Fig. (1). This is aware of directionality, thanks to the training, without directly using any of the deterministic algorithms.

Details of various elements of the overall algorithm are given below.

A. Sliding Window Algorithm

The first new method uses a hue and intensity sliding-window to calculate shadow regions. This looks at changes in regional variation between hue and intensity. The underlying colour model used here is HSV (hue, saturation, value), where hue is an 8-bit value. Sometimes methods that use super-pixel and sub-region sampling (such as CNNs) tend to be too localised; it is difficult to compare features of a region in an image with another, where global rather than local changes influence the distribution of segments and this can potentially result in underfitting inside a Neural Network training model. Therefore, our sliding window compares the value specific to the window with global mean-average trends of the image. The sliding window/image is calculated using Eq. (1) as follows:

$$S = \begin{cases} 1, & \text{if } \bar{H}_w > \bar{H}_i \text{ AND } \bar{V}_w > \bar{V}_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where S is a shadow candidate pixel, and its value is black (i.e. 1), if the mean hue of the window \bar{H}_w is greater than the mean hue of the image \bar{H}_i and the mean intensity of the

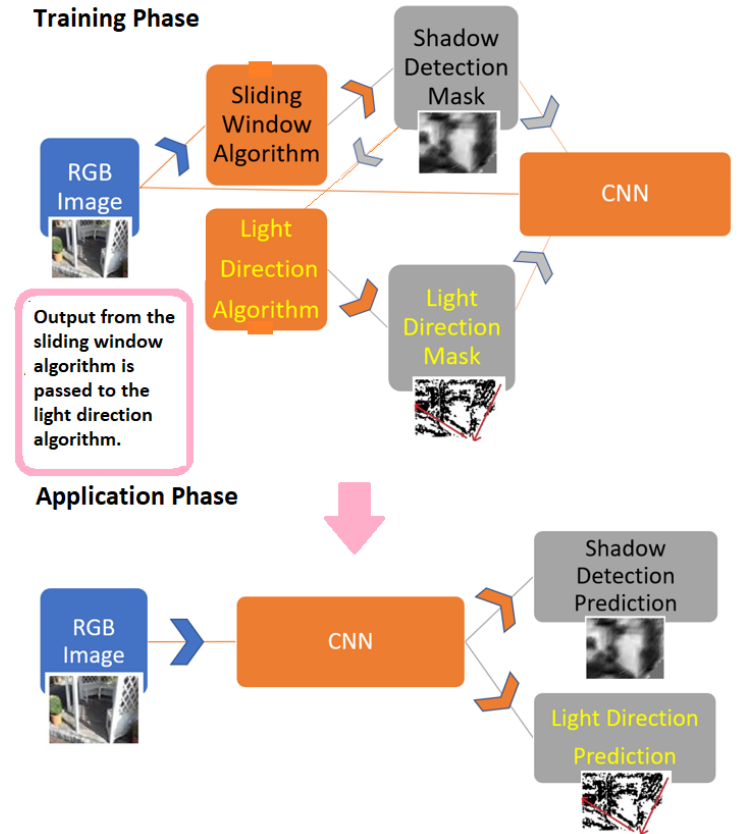


Fig. 1. The sequence of the algorithms during the training phase (top) and the application of the trained CNN (bottom).

window \bar{V}_w is greater than the mean intensity of the image \bar{V}_i , otherwise it is white (i.e. 0). The sliding window is set to dimensions of 11x11 pixels, with a centre pixel-value, from where the window moves. The mean i, j^{th} pixel-values of the window is calculated like so, for the hue and intensity of their respective positions:

$$\bar{H}_w = \frac{\sum_{i=1}^n \sum_{j=1}^n H_w i j}{n^2}, \quad \bar{V}_w = \frac{\sum_{i=1}^n \sum_{j=1}^n V_w i j}{n^2}. \quad (2)$$

The window values are then compared, at these positions, with global hue and intensity features, as noted in equation (2). The masks produced by this method are then fed to the CNN. This is the first significant difference from a conventional CNN. In a conventional CNN, the convolution windows are built into its architecture and cannot compare global to local values at specific positions, or calculate distances for the window in real-time, beyond super-pixel and sub-region sampling [35]. The code of our implementation is published in [30] in the function *slidingWindow* in *main.cpp*.

B. Light Direction Algorithm

The second method uses a modified Bresenham line algorithm [31]. The code of our implementation is published in [30] in the function *calcHistoData* in *main.cpp*. This provides a real-time geometry reference for light and shadow position; once these lines are drawn, the dominant

light-directions are passed to the CNN for training. The implemented Bresenham line algorithm plays a key role in the proposed neural system, since it is always able to return lines, with both positive or negative slopes, even in scenarios in which estimating the gradient might be difficult or mathematically impossible. The returned lines represent the directions giving the “orientation” functionality to the CNN, as it is trained with multi-directional real-time cues, rather than static cues, like those conventionally used in deep learning algorithms. In Fig. (2) and Fig. (3) a visual illustration of the method is provided.

The algorithm processes the output from the shadow detection algorithm step, as shown in Fig. (1). The shadow regions appear as black pixel zones in the output of the sliding-window algorithm. The principle behind the light-source detection is to determine the longest unbroken line of black pixels, starting from each position in the image, by looking into thirty-two directions. Our algorithm is a variation on the standard one in [32] in that it is not calculating the endpoints for *drawing* the line beyond the edge of the image, but it finds where the black pixel region ends. To that end, once the line reaches a white pixel, this causes it to stop and the loop continues, until all the pixels and lines have been calculated. The longest lines predict the largest gradients and the directionality of the light sources in the image.

The second major innovation on the Bresenham line algorithm is found in the number of gradients (32), used to do the line-reading calculation. A Bresenham line has to be specially configured to read/draw lines in several quadrants of orientation: without the lines simply being drawn with the same slopes facing into each other in different quadrants. 32, thus, was chosen to split the image into eight global zone-quadrants and have a further four zone-quadrants for the local position of the line. Since, on a pixel grid, a line which is not exactly horizontal or vertical must be approximated by pixels, which display steps, the 32 gradients were chosen so that the approximations of the lines cover the pixels of the image, but do not make the process computationally too expensive. This number was arrived at through trial and error, a smaller number of quadrants-within-quadrants (16), would not allow the lines to be generated with the capacity to cover the pixels of the image, whilst a larger multiple of quadrants-within-quadrants (64) became exponentially computationally expensive to calculate. We initially tested this by using a graphical tool to draw the individual lines, as the system was running, to be able to visually check what the lines were doing.

In summation, this study proposes an innovative use of the Bresenham Line-Algorithm, whose output is fed to a fully-connected layer of the Full CNN, being used to train the CNN with heuristic directional cues. Therefore, the light direction algorithm plays a key role in the proposed methodology as it produces the input to the second layer (mask) of the Full CNN, as noted in Fig 2.

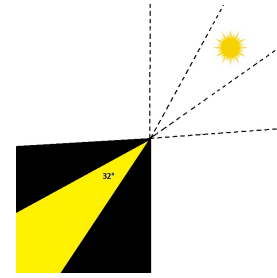


Fig. 2. A visual representation of the light source direction. The black pixels represent shadow lines, caused by obstructions of the light path, their angles, therefore, representing directionality of the light source. The yellow gradient (marked manually) corresponds approximately to the elevation of the Sun. Thirty-two degrees represents the span of the light source, lying between the two black-pixel shadow regions.

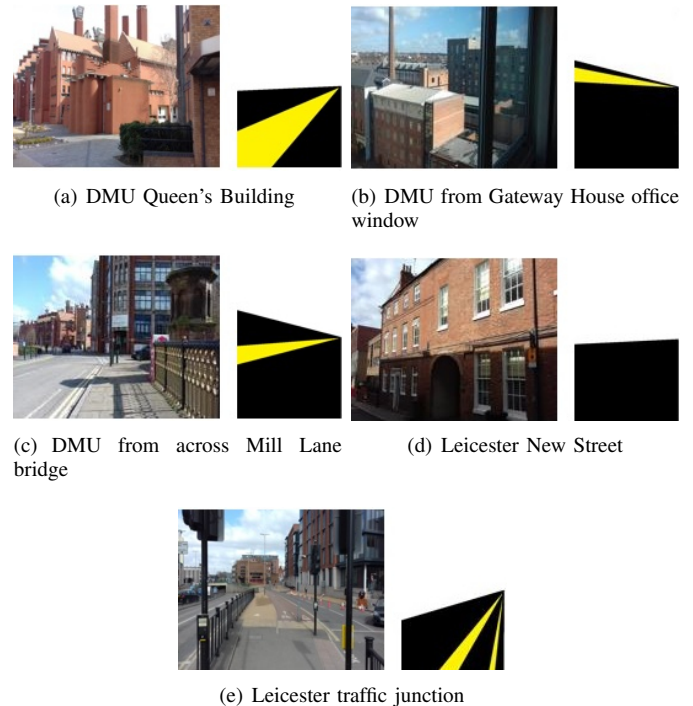


Fig. 3. Original images with corresponding light-direction algorithm output. In (e) the algorithm distinguishes between the sides on which the shadows fall (the shadow of the railing and the traffic light are facing into each other for example) and classifies them into two different gradients, on the same side. This is essential in the training of the full CNN.

C. Full CNN

The VGG-16 Network architecture was used as a basis for the Full CNN [33], though it was adapted significantly. The native filter size of 3x3 was changed to 32x32, arrived at through trial and error. Rectified Linear Units (RELU) [28] was the chosen activation function. This CNN, once trained as described, produces a shadow map and a light-direction estimation, as shown in the lower part of Fig. (1). Some typical results of the full CNN are shown in Fig. 4 and they will be analysed in Section (III). VGG-16 is considered one of the standard algorithms for image classification and segmentation. Whilst newer architectures are available, their improvements

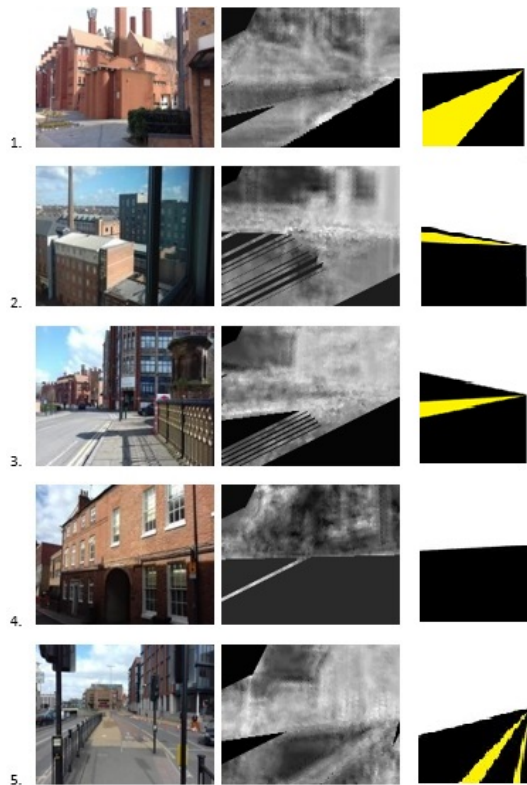


Fig. 4. A visual representation of the results of the full CNN. The second column shows the shadows detected while the third column the light direction. For the interpretation of the third column, see Fig. 2. An analysis of what the Full CNN is recognising in these images is given in Section III-B

are small and should not influence the concept of our methods. Moreover, RELU is the preeminent activation function in image classification and segmentation in deep learning [34], [35], [13], thus justifying their use here, in this context. A grid search approach has been used to determine the activation function (RELU, Softmax, Sigmoid etc.), and the number of Neurons per-layer. The number of training iterations was set to 600 for testing a combination of parameters. The structure of the proposed adapted CNN presents 32×32 convolutional filters and two main layers, one for the sliding-window input and one for the light-direction input respectively. The training process was performed in the CPU. Obviously, this must be taken into account when looking at the results of the testing, reported in Section III, in terms of both training and testing elapsed time (displayed in Table I). To strengthen the validity of the proposed approach, its accuracy was compared against the accuracy (indicated with α here) of two recent state-of-the-art methods, namely:

- the stacked-CNN in [36], here referred to as “Patch-CNN” (Table I);
- the adversarial-CNN method in [37], here referred to as “A&D-Nets” (Table I).

For the sake of clarity, it is worth stressing that non CNN-based light direction methodologies as e.g. those in [20], [21] have purposely been excluded from the comparison due to

their differing approaches, when compared against the CNN approach as explained in section I.

III. RESULTS

A. Performance of the Full CNN for Shadow Detection

In Fig. (4) and Fig. (5), the results of the light-direction step of the Full CNN are shown, in Fig. (6) the results of the shadow-classification step of the Full CNN are presented and in Fig. (7) the results of both the methods are shown and compared with two competing methods. For this, we determine how much of the shadow regions are correctly identified compared to the ground truth. The results of the light-direction step are analysed in Fig. (4) and (5). The results from the light-direction step of the Full CNN were binarised to produce the results seen in the second last row of Fig. (7). The second set of results are the output of the sliding-window (the shadow classification step), as seen in Fig. (6) and, as above, were binarised to produce the results of the third last row of Fig. (7). In Fig. (7) the full CNN and the results of the other competing deep learning methods are compared. The top image is the original RGB image, then the ground-truth; underneath the ground-truth, the other methods are included in the following order, for comparison: the method in [36], the method in [37], the output of the sliding window algorithm-step of the Full CNN and the light-direction step of the Full CNN. For calculating the accuracy of sliding-window step and the competing shadow classification methods, a method from ([37]) was employed, whereby the number of correctly guessed black-pixels was averaged with the number of the correctly guessed white-pixels (across the thirty images). It was found that using this metric the sliding-window output-step of the Full CNN had an accuracy of 80.7%, these results are reproduced in Table I. In the latter, the “total” accuracy is determined by means of Eq. 3:

$$\alpha_{\text{total}} = \frac{\alpha_S + \alpha_{N-S}}{2} \quad (3)$$

where the accuracy for the shadow regions “ α_S ” is calculated as the ratio of *black pixels* and *shadow pixels* while the accuracy for the non-shadow regions “ α_{N-S} ” is calculated as the ration of *white pixels* and *non-shadow pixels*.

For the results, a comparison between the number of correctly guessed black shadow regions to those of the ground-truth averaged with the number of correctly guessed white regions was used to evaluate accuracy. It was found that the full CNN compared favourably with the results of [37]. The new method had an approximate accuracy of 80.7%, standardised as compared with [37], considered the benchmark, with a difference of 15% in our favour (see Table I for an overview). The Full CNN did not share the same issues as the other methods with misclassifying glass and reflective surfaces (Fig. 7 D) and regions of solid colours, such as the traffic lights in the Leicester traffic junction image (Fig. 7 E). Since for example in Fig. 7 (C), the shadows on the street surface are correctly detected, it is clear that the neural network is not only picking up differences in the materials. In terms

of computational time, it took 1.7 seconds to process the set in [37], 11.1 seconds in the Full CNN, for the set of images and a projected 1.87 seconds for the set in [36].

Two additional evaluation metrics, IOU (aka the Jaccard method) and Sørensen-Dice coefficient are used on top of the previously defined B/W. The improvements here is less significant due to different calculation of the metrics.

B. Light-Source Direction Analysis

The light-direction algorithm output-step is not tested using the same metric as the sliding-window algorithm-step, even though they are both involved in the process. The light-direction algorithm step works by outputting consistent predictions of shadow regions along specific directional lines/gradients, rather than classifying regions, thus creating an indication of where light-sources originate, based on the accretion of these lines forming into solid-black pixel 'angles.' Instead of the pixel-region classification method, the output is visually compared with the calculated light-source direction. It can be seen in Fig. (4), Fig. (5) and Fig. (7) that the light-direction algorithm step can determine light-source directionality and, most importantly, it can determine light-sources coming from different directions simultaneously and assign weight to the distribution of these light-sources, if there are several. The authors have not yet attempted this using images from inside rooms, though presumably the same principles would apply in these circumstances.

Methods	B/W Accuracy	IOU	Dice	Time (sec)
Full CNN	80.70%	66.2%	75.4%	11.10 (CPU)
Patch-CNN [36]	58.10%	56%	72%	1.87 (GPU)
A&D-Nets [37]	64.40%	66%	75%	1.70% (GPU)

TABLE I

ACCURACY AND COMPUTATIONAL TIME: RESULTS AND COMPARISON OF THE FOUR METHODS. ACCURACY METRICS GIVEN IN *percent (%)* AND TIME IN *seconds (SEC)*. BEST ACCURACY AND ELAPSED TIME ARE MARKED IN BOLDFACE TODO WE NEED REFERENCES IN HERE.

IV. DISCUSSION

Two new contributions from this research have been discussed, in terms of the methods. The innovations from this research have applications in calculating and standardising directionality of normals in ray-tracing software, with the help of machine learning. The application would be able to do this by providing a standardised light-direction, based on actual physical light directionality, without the need for expensive rendering software, which is not currently feasible in game development and graphics pipeline applications. The second application is in creating 3D environments based on a photograph, without 3D scanning, or the need for several photographs being joined together. In this case, the data from the light-direction algorithm step of the Full CNN can output to a shader pipeline, creating a realistic geometry, based on real-time light and shadow position.

As noted in the Introduction it can also be an invaluable deep learning mechanism for machine vision, by providing a more directionally-accurate navigation system for autonomous

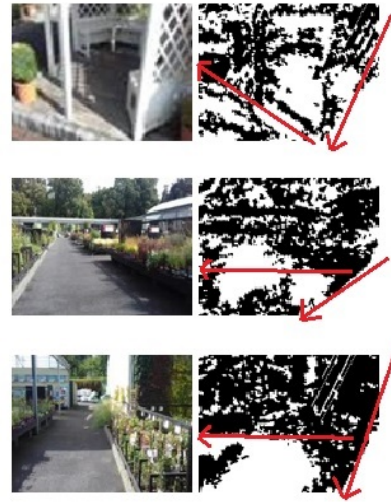


Fig. 5. RGB originals are on the left and the output is on the right. As can be seen in the above images, where red lines (added manually) indicate the light-direction ground-truth, the line-direction step of the Full CNN can interpolate light-source direction from multiple angles.

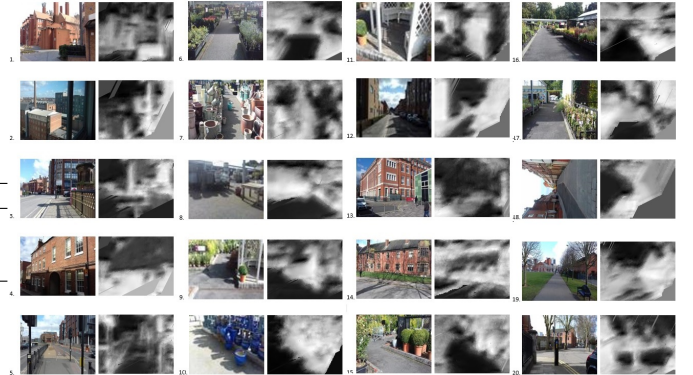


Fig. 6. Comparison of the original RGB images with the output of the shadow detection step of the Full CNN. Images were taken from Leicester city centre, DMU and a Garden Centre. Grey-pixel regions correspond to shadow regions in the above images, note that for clarity this order is reversed (black = shadows) in the results shown in the third last-row of Fig. (7). Some pictures were reversed and/or flipped, for a variety of training.

vehicles and drones, which currently rely on video and laser-based navigation-systems. To elaborate, convolutional neural networks cannot normally determine accurate directionality for their input layers concerning particular shapes, or regular geometric forms. This is the main use for the light-source detection algorithm. For example, in image classification, line segments can be passed as layers, but the Network is not learning about specific directionality, or orientation in a *semantic* sense. The new methodology should be able to improve the accuracy of existing methods used in autonomous vehicle and drone-navigation.

The proposed methodology offers a deterministic methodology to improve the training of unsupervised machine learning. It also allows a CNN to creatively and autonomously differen-



Fig. 7. Left to right: (A) DMU Queen’s Building image, (B) DMU from Gateway House office window, (C) DMU from across Mill Lane bridge and (D) Leicester New Street. Top to bottom: (1) original image, (2) ground-truth, (3) the adapted method in [36], (4) the method in [37], (5) the output of the sliding-window algorithm step of the Full CNN, (6) the output of the light-direction algorithm step of the Full CNN, and (7) the output of the deterministic algorithm for comparison. The first image (A) has light sources coming from two separate directions and shadows cast by a building partially visible in the image. The angle of this image (B) was chosen to assess whether or not a picture taken through glass affects the calculation of shadow regions and to create a contrast between the angle of projection of the sun and the sky. It was found that the sliding-window algorithm step of the Full CNN did not share the issues with misclassifying regions of glass (reflective surfaces), or areas of solid colour and hue, as the other methods (in (C) and (D)) [36]. Furthermore, the light-direction algorithm step of the Full CNN was able to classify light direction from two different directions (as in (A) and (B)).

tiate concepts, such as orientation or directional cues, which are currently lacking [38]. The traditional CNN approach relies on a stochastic learning approach which introduces often undesirable randomness, typically inherent in neural networks. Conversely, the aim of our methodology is not to provide larger, or more complex data sets for training, but rather to engineer data, which may be more granular and improve the training of a convolutional neural network.

V. CONCLUSION

In conclusion, this paper has looked at leveraging two novel methods in the training of a convolutional neural network, using deterministic methodologies. Unlike other approaches,

the authors proposed a blended-deterministic methodology, developed for detecting shadows to interpolate light-sources with this new CNN. The first of these methods is a sliding-window algorithm for shadow detection, the second innovation is a light-direction algorithm, which detects the direction of light sources in an image. Both of these methods were used to create input-layers to a fully-connected unsupervised CNN. The idea of comparing and potentially blending deterministic and stochastic models was discussed, to expand and enhance current machine learning research. Furthermore, the approach solves the problems of manually annotating large training data sets.

The new CNN has proven to be more accurate than competing methods in this area and, most crucially, has demonstrated the ability to predict light source-direction and thus orientation accurately, which is a significant breakthrough for an unsupervised CNN. This was possible also due to the innovative use of the line-direction algorithm in training the full CNN to determine directional cues from the light. To develop the methods further, we will expand on the results of this research by employing deterministic methods in the field of image processing and graphics, such as the sliding-window and Bresenham line algorithms, and applying them to current stochastic models used in the area of machine learning. Examples of where such tools would be useful are video and laser-based data employed in the navigation of autonomous vehicles, robots and drones, for example. We believe that using image processing and graphics methodologies can enhance the training of machine learning algorithms, specifically convolutional neural networks, which could prospectively use real-time graphical or visual cues, such as light-source interpolation, to supplement accuracy.

APPENDIX

The dataset of the original RGB images, their output of the sliding window step of the Full CNN and the light-source direction step presented in the left column of Fig. 3, first and second columns of Fig. 4, first column of Fig. 5, all columns of Fig. 6 and first, fifth and sixth rows of Fig. 7 can be found in [30]. This contains all training and test data used in this paper. Note that in Fig. 7 in rows five and six, the results were binarised and the colours were inverted (black=shadow regions, white=non-shadow regions) for clarity.

REFERENCES

- [1] G. Funke-Lea and R. Bajcsy, “Active color image analysis for recognizing shadows,” *Technical Reports (CIS)*, p. 307, 1992, technical Report No. MS-CIS-92-82.
- [2] R. Guo, Q. Dai, and D. Hoiem, “Single-image shadow detection and removal using paired regions,” in *CVPR 2011*. IEEE, June 2011, pp. 2033–2040.
- [3] M. K. Moghimi and H. Pourghassem, “Shadow detection based on combinations of hsv color space and orthogonal transformation in surveillance videos,” in *2014 Iranian Conference on Intelligent Systems (ICIS)*, Feb 2014, pp. 1–6.
- [4] A. Ozgun Ok, “Automated detection of buildings from single vhr multispectral images using shadow information and graph cuts,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 86, pp. 21 – 40, 2013.

- [5] T. F. Y. Vicente, M. Hoai, and D. Samaras, "Noisy label recovery for shadow detection in unfamiliar domains," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3783–3792.
- [6] M. D. Elkhoully, S. James, and A. Del Bue, "Multi-view aggregation for color naming with shadow detection and removal," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE, 2018, pp. 115–120.
- [7] S. H. Khan, M. Bennamoun, S. M., and T. R., "Automatic shadow detection and removal from a single image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 431–446, March 2016.
- [8] M. H. Panchal and N. C. Gamit, "A comprehensive survey on shadow detection techniques," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 2249–2253.
- [9] H. Mizes and V. Kozitsky, "Shadow detection and removal in license plate images," Jan. 11 2018, uS Patent App. 15/205,146.
- [10] S. Rees and M. Tschärke, "Methods, systems, and devices relating to shadow detection for real-time object identification," Mar. 28 2019, uS Patent App. 16/160,679.
- [11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
- [12] P. Liskowski and K. Krawiec, "Segmenting retinal blood vessels with deep neural networks," *IEEE Transactions on Medical Imaging*, vol. 35, no. 11, pp. 2369–2380, Nov 2016.
- [13] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, "Deep features learning for medical image analysis with convolutional autoencoder neural network," *IEEE Transactions on Big Data*, pp. 1–1, 2018.
- [14] T. F. Y. Vicente, L. Hou, C.-P. Yu, M. Hoai, and D. Samaras, "Large-scale training of shadow detectors with noisily-annotated shadow examples," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 816–832.
- [15] S. Mohajerani and P. Saeedi, "Cpnet: A context preserver convolutional neural network for detecting shadows in single rgb images," in *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2018, pp. 1–5.
- [16] N. Sugano, H. Kato, and K. Tachibana, "The effects of shadow representation of virtual objects in augmented reality," in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* IEEE, 2003, pp. 76–83.
- [17] Y. Liu and X. Granier, "Online tracking of outdoor lighting variations for augmented reality with moving cameras," *IEEE Transactions on visualization and computer graphics*, vol. 18, no. 4, pp. 573–580, 2012.
- [18] V. John, L. Zheming, and S. Mita, "Robust traffic light and arrow detection using optimal camera parameters and gps-based priors," in *2016 Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*. IEEE, 2016, pp. 204–208.
- [19] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.
- [20] J. Lopez-Moreno, S. Hadap, E. Reinhard, and D. Gutierrez, "Light source detection in photographs," in *Congreso Espanol de Informatica Grafica (CEIG)*, 2009, pp. 161–167.
- [21] A. Roy, S. Mitra, and R. Agrawal, "A novel method for detecting light source for digital images forensic," *Opto-Electronics Review*, vol. 19, no. 2, pp. 211–218, 2011.
- [22] G. Sun, H. Huang, Q. Weng, A. Zhang, X. Jia, J. Ren, L. Sun, and X. Chen, "Combinational shadow index for building shadow extraction in urban areas from sentinel-2a msi imagery," *International Journal of Applied Earth Observation and Geoinformation*, vol. 78, pp. 53 – 65, 2019.
- [23] T. Vicente, M. Hoai, and D. Samaras, "Leave-one-out kernel optimization for shadow detection and removal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 682–695, 2018.
- [24] G.-B. Lee, M.-J. Lee, W.-K. Lee, J.-H. Park, and T.-H. Kim, "Shadow detection based on regions of light sources for object extraction in nighttime video," *Sensors (Switzerland)*, vol. 17, no. 3, 2017.
- [25] M. Olano and D. Baker, "Lean mapping," in *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ser. I3D '10. New York, NY, USA: ACM, 2010, pp. 181–188.
- [26] Scratchapixel, "The phong model, introduction to the concepts of shader, reflection models and brdf;" www.scratchapixel.com/lessons/3d-basic-rendering/phong-shader-BRDF, 2019, online; accessed 6 June 2019.
- [27] E. Gawehn, J. A. Hiss, J. B. Brown, and G. Schneider, "Advancing drug discovery via gpu-based deep learning," *Expert Opinion on Drug Discovery*, vol. 13, no. 7, pp. 579–582, 2018, PMID: 29668343. [Online]. Available: <https://doi.org/10.1080/17460441.2018.1465407>
- [28] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354 – 377, 2018.
- [29] X. Li, H. Chen, X. Qi, Q. Dou, C. Fu, and P. Heng, "H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes," *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, pp. 2663–2674, Dec 2018.
- [30] S. Colreavy-Donnelly, S. Kuhn, S. O'Connor, F. Caraffini, and Z. Anastassi, "Light sources & shadow detection dataset," <http://dx.doi.org/10.17632/4fgys3cj58.3>, 2019, mendeley Data, v3.
- [31] M. Abrash, *Michael Abrash's Graphics Programming Black Book, with CD: The Complete Works of Graphics Master, Michael Abrash*, 10th ed. Scottsdale, AZ, USA: Coriolis Group Books, 1997.
- [32] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [33] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.
- [34] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 2146–2153.
- [35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2999134.2999257>
- [36] S. Hosseinzadeh, M. Shakeri, and H. Zhang, "Fast shadow detection from a single image using a patched convolutional neural network," *CoRR*, vol. abs/1709.09283, 2017. [Online]. Available: <http://arxiv.org/abs/1709.09283>
- [37] H. Le, T. F. Y. Vicente, V. Nguyen, M. Hoai, and D. Samaras, "A+d net: Training a shadow detector with adversarial shadow attenuation," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 680–696.
- [38] G. Barna and K. Kaski, "Stochastic versus deterministic neural networks for pattern recognition," *Physica Scripta*, vol. T33, pp. 110–115, 1990.