
Interpretable deep learning for guided structure-property explorations in photovoltaics

Balaji Sessa Sarath Pokuri

Department of Mechanical Engineering
Iowa State University
Ames, IA 50011
balajip@iastate.edu

Sambuddha Ghosal

Department of Mechanical Engineering,
Department of Computer Science
Iowa State University
Ames, IA 50011
sghosal@iastate.edu

Apurva Kokate

Department of Computer Science,
Iowa State University
Ames, IA 50011
akokate@iastate.edu

Baskar Ganapathysubramanian

Department of Mechanical Engineering
Iowa State University
Ames, IA 50011
baskarg@iastate.edu

Soumik Sarkar

Department of Mechanical Engineering
Iowa State University
Ames, IA 50011
soumiks@iastate.edu

Abstract

The performance of an organic photovoltaic device is intricately connected to its active layer morphology. This connection between the active layer and device performance is very expensive to evaluate, either experimentally or computationally. Hence, designing morphologies to achieve higher performances is non-trivial and often intractable. To solve this, we first introduce a deep convolutional neural network (CNN) architecture that can serve as a fast and robust surrogate for the complex structure-property map. Several tests were performed to gain *trust* in this trained model. Then, we utilize this fast framework to perform robust microstructural design to enhance device performance.

1 Introduction

Mapping microstructure to macro-scale property of materials and controlling such relationships has been an important theme of modern materials research [11]. Despite the investment of millions of research hours and dollars [1], this largely remains elusive to the research community. A major reason is that the map from microstructure (domain) to property (co-domain) is inherently non-surjective and highly non-linear, making incremental judgments based on incremental modifications infeasible. Added to this is the infinite dimensionality of microstructure space, which makes the description of the domain in-exhaustible. Hence, estimating the property of one morphology from another visually similar morphology is not trivial – each morphology will require a full scale experiment/simulation for reliable quantification. In this paper, we discuss one such problem and present a solution using modern machine learning paradigms. More specifically, we look at solving the relationship between active layer microstructure and performance of an Organic Solar Cell (OSC). OSCs utilize a bulk heterojunction morphology in the active layer to efficiently convert incident solar radiation into

electrical energy [19]. The total power conversion efficiency is very intricately related to this material distribution if acceptor and donor polymers in the active layer (See Sec. 2). This relationship has several contradicting and competing factors. For example, larger interfacial area between acceptor and donor regions enables higher production of charges, but simultaneously also increases the loss of charges through recombination. While larger domains can help transport charges faster, they also impede excitons from dissociating into charges. These contradictory features make this an exciting area of research, finally culminating in the search for the most efficient microstructures for solar energy conversion.

Traditionally, this structure-property map was computationally resolved either through a detailed and tedious PDE simulation [14] or Kinetic Monte Carlo simulation [4, 16, 15, 21] or through identification of surrogate descriptors of performance [22]. While the first method is robust and accurate, it is limited by the size of microstructure. Often it requires massive, robust, and well established computational resources along with a perfectly scaling computational model to solve for even simple representations of morphologies. Hence such analysis was often limited to 2D microstructure quantification, although 3D quantification is not uncommon [14]. The second method overcomes most of these computational challenges. Typically intuitive features of the microstructure such as domain sizes and orientations, connectivities and islands are quantified (using graphs) instead of performing a full scale PDE simulation. This technique is intuitive and requires substantially less computational resources (although memory requirements are higher). Subsequently, it enables simultaneous parallel analysis of multiple morphologies at once. Moreover, close to linear dependence of several stage efficiencies on these microstructural descriptors was also shown [23]. However, since the dimensionality of the domain is infinite, the intuitive metrics tend to be a limited description of the bigger reality.

In this work, we take the approach called DLSP (Deep Learning based Structure-Property exploration) involving machine learning to solve this complex map of microstructure to performance (depicted in Fig. 1). Modern machine learning has been shown to be a powerful tool to learn complex phenomena [6], especially image recognition [24, 2]. This technique has the potential to identify features in the microstructure that are not intuitive to the simple eye. By posing the task of enumerating microstructure informed performance as an image recognition problem, we can easily and confidently extend current implementations of neural networks to this problem. Furthermore, interpreting the trained network can give both trust in training as well as insights into the most impactful features of a microstructure, thereby enabling researchers to tailor processes to create such features.

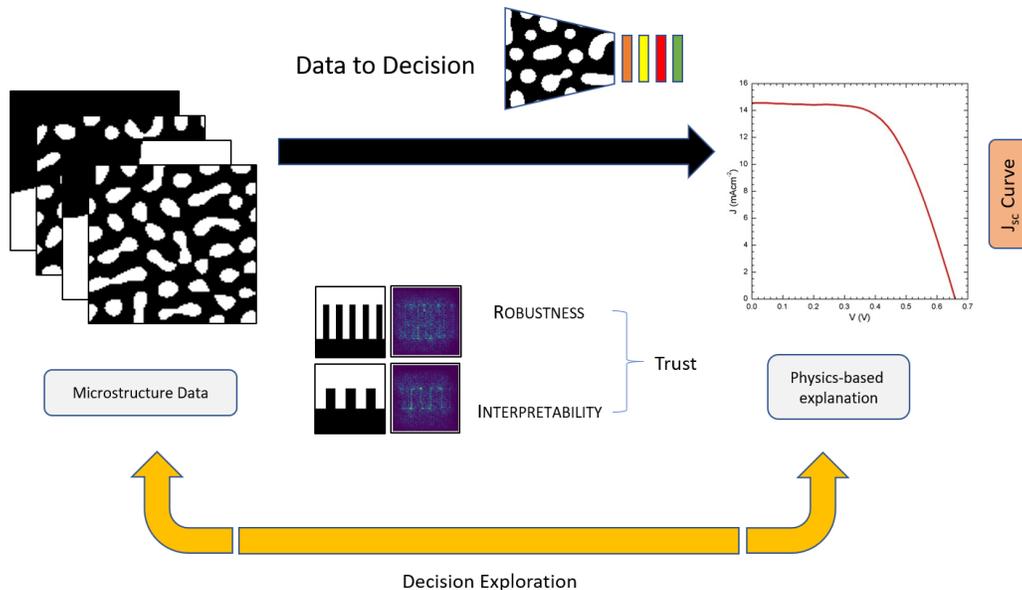


Figure 1: DLSP framework: We construct a forward map from morphology to performance. Upon building *trust* in this trained model, it was further used for performing manual and automated design.

The rest of the paper is organised as follows: In Sec. 2, we discuss the physics behind the problem of morphology design. Subsequently, a short discussion on the methods for data generation and labelling is discussed. The details of network architecture and other machine learning parameters are discussed in Sec. 3. In Sec. 4, the results of training, validation and testing on in-sample and out-of-sample morphologies and performance interpretations are presented. The benefits of such fast interpretable forward surrogate model are demonstrated through the design of property-maximized-microstructure in Sec. 5. Finally in Sec. 6, key takeaways from this work and future directions are discussed.

2 Physics of structure-property explorations in photovoltaics

2.1 Organic Photovoltaics

Organic photovoltaic devices are energy harvesting devices which employ organic materials for solar energy conversion. These provide multiple advantages over traditional silicon based cells, like flexibility, transparency and ease of manufacturability. They however are limited by their efficiency of operation. Although major breakthroughs in processing and materials have improved the efficiency drastically, they still lag behind the traditional photovoltaics.

The efficiency of these devices is intricately dependant on the material distribution/morphology in the active layer. The active layer generally is a bulk hetero-junction, enabling multiple sites for charge generation. Several features of the morphology have different roles in the process of converting solar energy. The ability to change these morphological features by changing the processing protocol is a major source of control in these devices.

There are several stages during the solar power conversion in an OPV. Firstly, the incident solar energy generates excitons in the donor phase. These excitons are highly unstable and need to diffuse to a nearest interface with the acceptor material to separate into positive and negative charges. This diffusion to the interface is critical to evaluate the efficiency of absorption of incident light. The dissociation of excitons to form charges depends on the nature of interface and the materials in the interface. For example, interfaces with non-aligned crystal boundaries show lower dissociation than those with aligned crystals. In the next stage, these charges (positive charge in the donor and negative charge in the acceptor) need to be drifted to the respective electrode to produce electricity. Usually, this drift is provided by the potential difference between the two electrodes. However, these charges also encounter other interfaces which have pairs of positive and negative charges, leading to potential recombination. In summary, the total efficiency of the active layer involves exciton production, charge separation and charge transportation efficiencies.

In this context, quantifying the dependence of the device as well as stage efficiencies becomes a critical part in developing strategies to design processing conditions. It can already be seen that the role of morphology cannot be over-estimated in the power conversion efficiency. Hence strategies were developed to quantify the efficiencies these morphologies. While these techniques are robust and rigorous, they are expensive and time intensive. This makes them infeasible for further designing morphologies, which often requires several quantifications. So, we turn to modern fast methods of quantifying data, especially images. We represent the morphologies as images and take advantage of the deep convolutional neural networks to do performance based classification.

2.2 Data generation and quantification

In order to train the network (to be described in Sec. 3), we generate a dataset of microstructural images using the Cahn-Hilliard equation. This process generates time series of images that can be treated as independent images for the sake of training. This method helps to quickly produce several thousands of images within a very short amount of time. Previous analysis using these images can be found in [23]. A characteristic of this procedure for generating morphologies through simulation is their similarity to morphologies in real active layers produced during thermal annealing. In the morphology, the domains are similar in size and have smooth interface contours. These characteristics will also help us to build trust in the training process by manually create morphologies breaking these characteristics and testing the performance of the trained network. More implications of this will be in discussed in Sec. 4. Using data augmentation techniques over the originally simulated data, we finally produce a dataset of nearly 65,000 (2D) morphologies. Each morphology is a gray-scale image of size $101\text{px} \times 101\text{px}$.

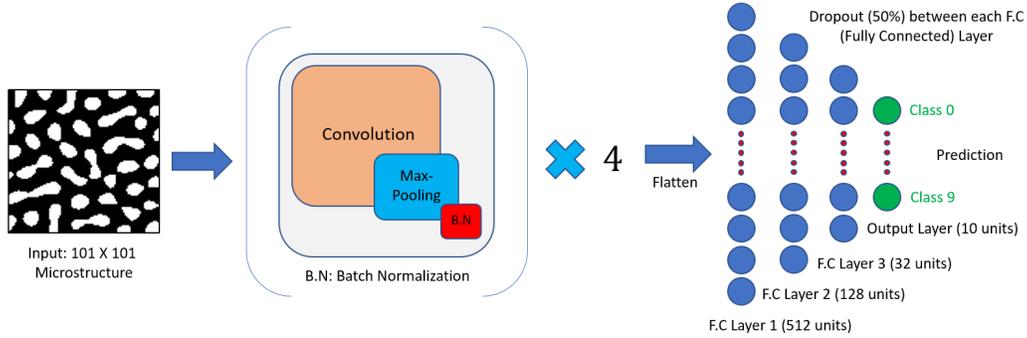


Figure 2: Proposed CNN architecture. Note that this is much shallower and has less trainable parameters compared to VGG-16 and ResNet-50.

These morphologies were then characterized using an in-house physics based simulator [14]. This simulator uses steady state excitonic drift diffusion equation to model the processes of exciton dissociation and charge transport. We use the short circuit current J_{sc} as a means of labelling the data. The whole data was divided into 10 classes which are equally spaced between the best ($J_{sc} = 7mA/cm^2$) and worst performing ($J_{sc} = 0.2mA/cm^2$) in the data.

3 Proposed Deep Learning framework

Convolutional Neural Networks (CNN) are well-established architectures for classification of images. They have proved to perform outstandingly well in terms of efficiently extracting complex features from images and function as a classification technique when provided with sufficient data [6, 24, 2, 17, 20, 8, 7]. Hence, to address the task above, we develop a CNN based architecture to classify the morphologies. The data label, originally a continuous value, is discretized into 10 bins. The network architecture is depicted in Fig. 2, with 1.2 million learning parameters. This architecture was taken from [3], where the original aim was to determine steering angles for a car from the dash camera image. The task is similar here also – both use images from time series data with predefined label to train. Our network is slightly shallower, to enable better training, enhanced with better feature visualization. This is also to prevent problems that can arise with deeper networks – vanishing (or exploding) gradients [9], which provide a hindrance to convergence, and the accuracy saturation with increasing depth. Shallower networks also provide better feature visualizations – in this case, we use saliency maps [12] to visualize learnt features (Sec. 4). The network was trained for approximately 70 epochs (with 18s per epoch) with a learning rate of 0.0001, on the 45,000-image training set to reach the desired accuracy (95.41%). The crossentropy (categorical) loss (or cost) function along with the Adam optimizer [13] was used to minimize the error.

Apart from this, we also tested two standard architectures with our dataset:

- VGG-16 (learning parameters 50 million), with learning rate of 0.0001, batch size of 128 initialized with random weights was also trained on the training dataset, achieving a test accuracy of 96.61% at epoch 70 (with 180s per epoch).
- ResNet-50 (learning parameters 23 million), with learning rate of 0.0001, batch size of 128 initialized with random weights was also trained on the training dataset, achieving a test accuracy of 96.45% at epoch 70 (with 580s per epoch).

We can see that our custom (shallower) CNN, while with lower training parameters, performs very similar to the established deep CNNs. Therefore, we defer the choice of selecting a "good" network to be based on the learnt features and out-of-sample performance and not just the accuracy/f1-score of the testing dataset.

4 Results

Figure 3 shows the confusion matrices for in-sample test data classification. It had an accuracy of 95.41% and F1-score of 94.45%. From the confusion matrix, it can clearly be seen that most of the classification is correct, and those which are wrongly predicted are only off by one class. This wrong prediction is not unexpected, as we are binning a continuous variable into non-overlapping classes. The edge cases have the potential to be misclassified. It is also to be noted that other two standard architectures show similar confusion matrices, with similar prediction accuracies.

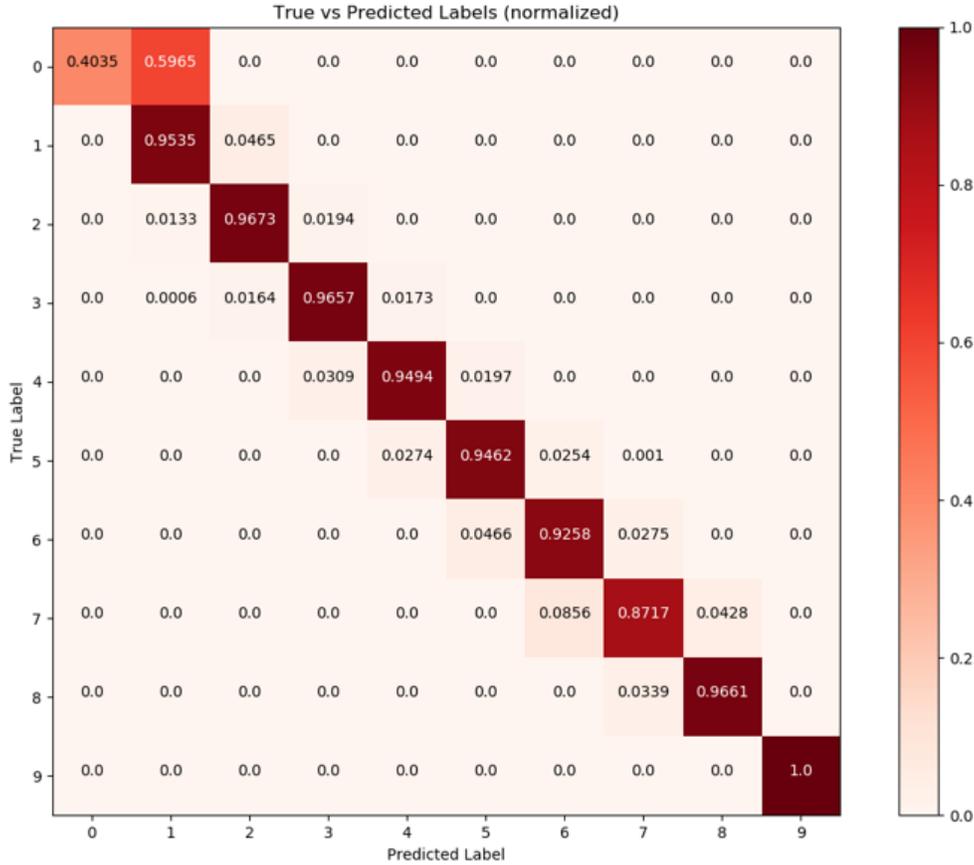


Figure 3: Confusion matrix for insample test predictions. Notice the heavily diagonally dominant matrix, indicating a very good classification accuracy.

4.1 Out-of-sample testing

While it is a known fact in the machine learning community that neural networks in their training phase can possibly overfit (depending on the model capacity, amount of training data and training hyperparameters), we resort to two methods of checking the robustness of our trained network(s). As noted earlier, the morphology data used for training is generated by solving a PDE. This inherits certain properties to the data such as smooth contours and uniform domain sizes. Hence we try to systematically break these assumptions about the dataset and see the performance of the network. Firstly, we test the network on a columnar structure (Fig. 4b). This is a widely studied structure, often considered as a high performing morphology. This is a completely out-of-sample data. This columnar morphology also has several sharp interface contours, which are completely absent in the training dataset. The results are in Fig. 4b. The actual J_{sc} values from a full scale drift-diffusion simulation (along with the corresponding true label) are also presented. We can see how the network is able to predict the correct label corresponding to each of the columnar microstructures.

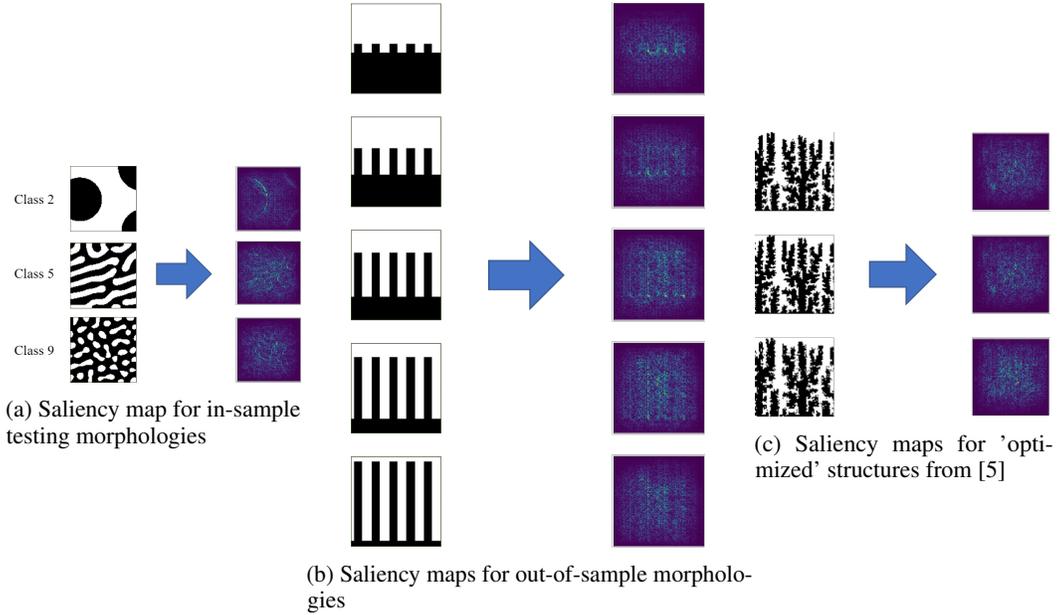


Figure 4: Saliency maps and performance of our custom trained CNN. Note how the saliency maps closely follow the interface regions in the microstructure. It should also be noted that the networks shows good performance even on samples outside the training dataset.

Next, we also test our framework with the "high" performing morphologies described in [5]. These morphologies were obtained through optimization strategies using graph based surrogate performance descriptors. We can also see how our network has rightly identified (Fig. 4c) *all* these as high performing class label 9.

4.2 Interpretability

We next query the network to understand the learnt features. For this, we use saliency maps [12, 18] to identify important features of the image input. Saliency map visualization is a visualization technique that generates heat maps on test images that bring out (highlight) the regions (microstructure regions, for our case) the trained CNN model focuses on to generate it's classification output. The heat-map signifies the aspect that regions highlighted more than the others carry a greater significance towards the model's classification output. Fig. 4 shows the saliency maps for morphologies in the data, columnar structures and the "high" performing morphologies identified in [5].

We can see, in all Figs. 4a, 4b & 4c, how the network highlights interface between the acceptor and donor regions. This is a vital test of the learning capabilities of the network. In terms of the underlying physics, the interface is the most critical feature of performance. Properly configured interface enables charge separation and hence better conversion. Poorly configured interface enables poor dissociation and higher recombination, which depreciates conversion efficiency. Finally, interface further away from the top and bottom electrodes are more critical, as the charges produced at these locations have a higher chance of recombination. We can see from Fig. 4 how the network is able to find the right type of interface as critical to device performance.

Furthermore, we also observe from Fig. 5 that the saliency maps from the standard deep networks (VGG-16 and ResNet-50) are unable to locate these above features. Although, the test accuracy is slightly higher than our custom network, we see that the saliency outputs do not provide us with any understandable information. This observation is in line with [10], where it was shown that deeper models are harder to explain than their shallower counterparts even though they may achieve a higher classification accuracy. These results signify the importance of tailoring architectures to the application. Thus, for performing morphology design, we use the custom architecture as a surrogate map from the microstructure space to the performance space.

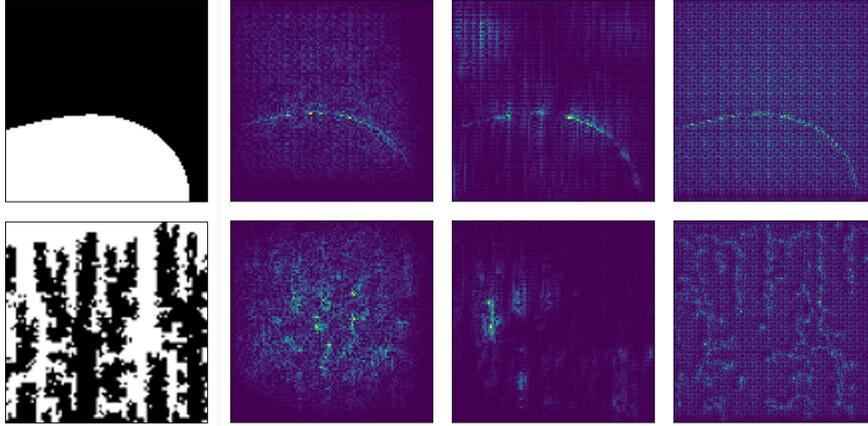


Figure 5: Comparison of Saliency map outputs for our Custom Model (second column), VGG-net (third column) and ResNet-50 (fourth column), with input images shown in the first column: top row shows an example image for class 0, bottom row shows an example image from omega morphologies (correctly predicted as class 9 by our custom model)

5 Morphology design

Having developed a fast and *trust-worthy* surrogate map from the microstructures to the performance, we can use it enable microstructural design. In this section, we show two separate techniques, manual and automated, for designing microstructures. The goal of both these techniques is to explore (uncanny) morphologies that demonstrate superior performance. Traditionally, this was generally achieved through a conventional optimization strategy, like simulated annealing, where an initial morphology is tweaked repeatedly to achieve superior performance. At every stage, the current morphology is evaluated for its performance. Subsequently, the whole process requires several computationally expensive evaluations and hence becomes time consuming. But, with the above demonstrated framework, evaluating the morphology becomes significantly faster and easier. Hence it provides an very powerful way to quickly 'evolve' morphologies to optimize performance.

5.1 Manual design

In order to enable manual exploration, we created a browser interface (Fig. 6) that enables the user to interactively modify morphologies to both visualize and improve morphology performance. Using this interface, the user can incrementally add changes to the initial morphology that can improve the predicted performance. Since the performance assessment is done by the trained CNN, the whole process happens real-time. Fig. 7 shows how one can modify images to include several features of varying sizes, with the aim of improving performance. This tool can in turn help identify features of morphology that affect the performance.



Figure 6: Browser interface for performing manual exploration and design.

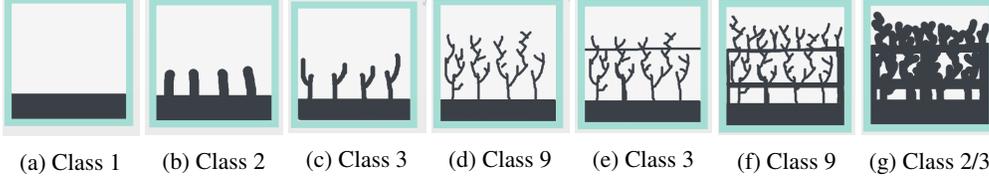


Figure 7: Exploration by manual design: Notice how the trained network captures the underlying physics – addition of non-interfering features improves performance. When a ‘blocking’ layer was added in (e), the performance drops. Also note that increasing domain size (in (g)) leads to performance drop, in conjunction with reduced exciton dissociation efficiency and increased recombination.

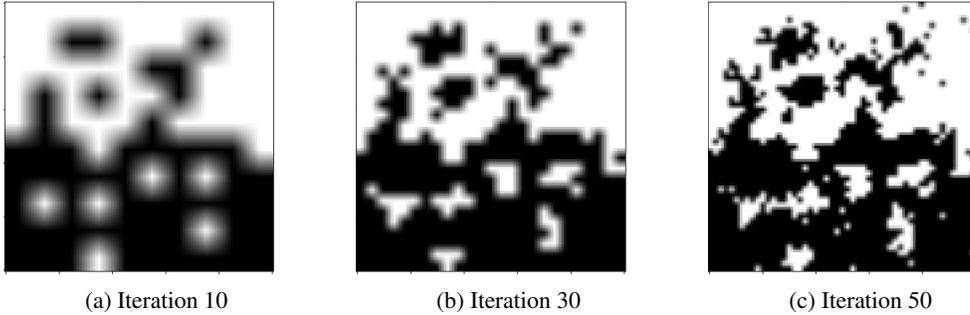


Figure 8: Exploration by automated design: The optimization started with a bilayer structure. Notice how the framework directs the formation of finer features

5.2 Automated design

While the above interface may serve very well to understand the influence of morphological features on performance, it is not feasible to perform a full scale morphology design. Manual exploration limits the full exploration of the *best performing morphology manifold*. Thus, to fully explore this space, we link this fast surrogate with a probabilistic optimization algorithm to find those optimal structures. More specifically we use a population based incremental learning (PBIL) approach to model morphologies and evolve them to achieve optimum performance. PBIL estimates the explicit probability distribution of the optimal morphology. The multi variate probability distribution is stored as a probability matrix P of the 2D morphology, i.e., each pixel is associated with a probability. This matrix P is updated as follows: the optimization starts with a given probability matrix, generally based on the intuition of the researcher. Subsequently, n morphology instances are sampled around this matrix P . For each realization, the DLSP is deployed to evaluate the performance, $f_j, j \in [1, n]$. Then n_b best samples ($n_b < n$) are used to calculate, P_u , the probabilistic update matrix. Next, the probability vector is updated according to $P = P \cdot (1 - l_r) + P_u \cdot l_r$, where l_r is the learning rate. Intuitively, the update step reinforces features present in the best performing morphologies, and dampens those missing. The algorithm terminates by standard criteria (iteration limits and improvement bounds). Only the probability matrix is stored and multiple realizations’ evaluations can embarrassingly parallel.

The results of PBIL optimization are very promising. Fig. 8 indicates structure at different scales, mimicking finger-like fractal structures. These are very similar to the results presented in [5].

6 Conclusion

In this work, we address the issue of designing active layer morphologies to enhance device performance, especially in OPV applications. The usual methods to quantify morphology are either too costly or too simplistic. Hence we take a data-driven approach (DLSP) to create a *morphology quantifier* that can perform fast evaluations. We train a custom designed CNN that reads the morphology and classifies it into 10 bins of increasing performance metric J_{sc} . Using out-of-sample datasets, we confirm that there is no severe over-fitting issues during the training process. Two other standard

networks (VGG-16 and ResNet-50) were trained end-to-end independently. It was observed that the custom network, although shallower, gave very similar accuracy. However, our custom network performed much better when visualized using saliency maps as well as when tested on out-of-sample datasets. It identified critical features of the interface in the morphology, which both VGG-16 and ResNet-50 failed to identify consistently. The custom designed network is then used to perform morphology design for achieving enhanced performance. Two approaches were taken to do this – the first one aims to inform the user about the effect of morphology on performance. The second approach uses the *trust-worthy* network as a fast cost function and performs morphology optimization using PBIL algorithm. It should be noted that this work serves as a proof of concept of using deep neural networks for material morphology quantification. It raises several other interesting questions of how to integrate physical phenomena into the training process. Can these physics based intuitions can be exploited to reduce the demand on the size of data for training? Can a most effective dataset be created to reduce the training data? Can we make the training process more robust to adversarial attacks? All these questions form the scope of future study.

References

- [1] Designing materials to revolutionize and engineer our future. https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=505073. Accessed: 2018-11-01.
- [2] B. Alipanahi A. Delong, M.T. Weirauch and B.J. Frey. Predicting the sequence specificities of dna-and rna-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838.
- [3] 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.
- [4] M. Casalegno, G. Raos, and R. Po. Methodological assessment of kinetic monte carlo simulations of organic photovoltaic devices: The treatment of electrostatic interactions. *The Journal of chemical physics*, 132(9):094705, 2010.
- [5] P. Du, A. Zebrowski, J. Zola, B. Ganapathysubramanian, and O. Wodo. Microstructure design using graphs. *npj Computational Materials*, 4(1):50, 2018.
- [6] A. Esteva. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.
- [7] S. Ghosal, A. Akintayo, P. Boor, and S. Sarkar. High speed video-based health monitoring using 3d deep learning. *Dynamic Data-Driven Application Systems (DDDAS)*, 2017.
- [8] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar. An explainable deep machine vision framework for plant stress phenotyping. *Proceedings of the National Academy of Sciences*, 115(18):4613–4618, 2018.
- [9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *AISTATS*, 2010.
- [10] A. Nguyen T. Fuchs J. Yosinski, J. Clune and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint*, 1506.06579, 2015.
- [11] S. Ju, T. Shiga, L. Feng, Z. Hou, K. Tsuda, and J. Shiomi. Designing nanostructures for phonon transport via bayesian optimization. *Physical Review X*, 7(2):021024, 2017.
- [12] A. Vedaldi K. Simonyan and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint*, 1312.6034, 2013.
- [13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 1412.6980, 2014.
- [14] H. K. Kodali and B. Ganapathysubramanian. Computer simulation of heterogeneous polymer photovoltaic devices. *Modelling and Simulation in Materials Science and Engineering*, 20(3):035015, 2012.

- [15] R.A Marsh, C. Groves, and N. C. Greenham. A microscopic model for the behavior of nanostructured organic photovoltaic devices. *Journal of applied physics*, 101(8):083509, 2007.
- [16] L. Meng, Y. Shang, Q. Li, Y. Li, X. Zhan, Z. Shuai, R. GE. Kimber, and A. B Walker. Dynamic monte carlo simulation for highly efficient polymer blend photovoltaics. *The Journal of Physical Chemistry B*, 114(1):36–41, 2009.
- [17] V. Mnih. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [18] G. Montavon, W. Samek, and K. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.
- [19] B. Ray, M. S. Lundstrom, and M. A. Alam. Can morphology tailoring improve the open circuit voltage of organic solar cells? *Applied Physics Letters*, 100(1):7, 2012.
- [20] D. Silver. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [21] P. K. Watkins, A. B. Walker, and G. LB Verschoor. Dynamical monte carlo modelling of organic solar cells: The dependence of internal quantum efficiency on morphology. *Nano letters*, 5(9):1814–1818, 2005.
- [22] O. Wodo, S. Tirthapura, S. Chaudhary, and B. Ganapathysubramanian. A graph-based formulation for computational characterization of bulk heterojunction morphology. *Organic Electronics*, 13(6):1105–1113, 2012.
- [23] O. Wodo, J. Zola, B. S. S. Pokuri, P. Du, and B. Ganapathysubramanian. Automated, high throughput exploration of process–structure–property relationships using the mapreduce paradigm. *Materials discovery*, 1:21–28, 2015.
- [24] D.L. Yamins and J.J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356, 2016.