npj | Computational Materials

www.nature.com/npjcompumats

ARTICLE OPEN

Interpretable deep learning for guided microstructure-property explorations in photovoltaics

Balaji Sesha Sarath Pokuri[1,3], Sambuddha Ghosal[1,2,3], Apurva Kokate[2], Soumik Sarkar[1,2]* and Baskar Ganapathysubramanian[1]*

The microstructure determines the photovoltaic performance of a thin film organic semiconductor film. The relationship between microstructure and performance is usually highly non-linear and expensive to evaluate, thus making microstructure optimization challenging. Here, we show a data-driven approach for mapping the microstructure to photovoltaic performance using deep convolutional neural networks. We characterize this approach in terms of two critical metrics, its generalizability (has it learnt a reasonable map?), and its intepretability (can it produce meaningful microstructure characteristics that influence its prediction?). A surrogate model that exhibits these two features of generalizability and intepretability is particularly useful for subsequent design exploration. We illustrate this by using the surrogate model for both manual exploration (that verifies known domain insight) as well as automated microstructure optimization. We envision such approaches to be widely applicable to a wide variety of microstructure-sensitive design problems.

npj Computational Materials (2019)5:95 ; https://doi.org/10.1038/s41524-019-0231-y

INTRODUCTION

Modern engineering applications are driving the demand for heterogeneous materials with tailored multifunctional properties. Very often, these properties are dependent on the microstructure. In recent years, there has been a sustained focus on microstructure-sensitive design. The design intent here is to identify tailored microstructures that result in desired properties.

The rational design of heterogenous materials has emerged as a very promising approach towards discovery of new materials and devices with tailored properties and subsequently spur novel applications. One such application example has been that of organic electronics, specifically organic photovoltaics (OPV). In spite of exhibiting multiple benefits (tunability, flexibility, cost, low-temperature manufacturability), organic photovoltaic films still remain a niche market due to relatively poor photoconversion efficiency compared to inorganic counterparts. Careful theoretical[1–6] and experimental analysis[7–10] have revealed how the microstructure impacts each stage of the photoconversion process. However, the complexity of these analysis approaches have made systematic exploration infeasible, with the result that there exist no design principles nor approaches for identifying promising microstructure in a systematic way. Thus, a key bottleneck to microstructure-sensitive design is the paucity of techniques that can rapidly evaluate the performance of a microstructure.

Our approach to resolve this bottleneck is through machine learning (ML), which is used to create a fast surrogate for any complex functional map in a data-driven manner. Over the last decade, machine learning models have proved their ability to ingest volumes of data-label pairs and create efficient proxy or surrogate models to predict labels for similar instances of data. Deep Learning, the state-of-the-art ML form, has especially advanced the field by incorporating the ability to learn features from high-dimensional data such as multi-spectral images,[11–13] speech[14] and text.[15] A particular form of deep networks called Convolutional Neural Networks (CNN) has become very popular due to its ability to autonomously create and analyze features in image-like inputs. Through the use of convolution operations, these models retain spatial neighbourhood information, thus allowing linking local (hierarchical) features of an image and an associated label, without the need for hand crafting of any features. Due to this special ability of ML algorithms to be input agnostic, i.e., the ability to automatically evaluate features from input data, they have found utility in a wide variety of applications including recommendation systems[16] and self-driving cars.[17] These approaches are slowly gaining popularity in physics and engineered systems,[18–20] where modern sensor and computational developments have paved the way for structured data generation.[21,22]

Here, we utilize the versatility of CNNs to map the active layer morphology of thin film OPVs to a performance metric, which is the short-circuit current J_{sc}. Specifically, we train a morphology classifier that maps a OPV morphology to a short-circuit current. We test several architectures (of varying depth and width) that can learn from a given set of morphologies and their labels, and demonstrate very high accuracy, and *F*1 score. To distinguish and rank order between these equally well performing models, we used two additional measures. The first is based on the observation that a good model must be able to generalize the learnt structure-property relationship. Thus, we identify network architectures that can generalize the map with the available dataset. We quantify this in terms of the ability of the architecture to 'project the unseen' morphology onto the learnt distribution and make good predictions.

Apart from generalizability, the other critical requirement for the ML model in our context is interpretability. While model interpretability is not a very critical metric for some applications (for instance, network failure or stock pricing), it becomes a fairly important metric for understanding the behavior of engineered systems. This is because having a purely predictive 'black-box'

[1]Department of Mechanical Engineering, Iowa State University, Ames, IA, USA. [2]Department of Computer Science, Iowa State University, Ames, IA, USA. [3]These authors contributed equally: Balaji Sesha Sarath Pokuri, Sambuddha Ghosal. *email: soumiks@iastate.edu or B.G.baskarg@iastate.edu

Published in partnership with the Shanghai Institute of Ceramics of the Chinese Academy of Sciences

npj nature partner journals

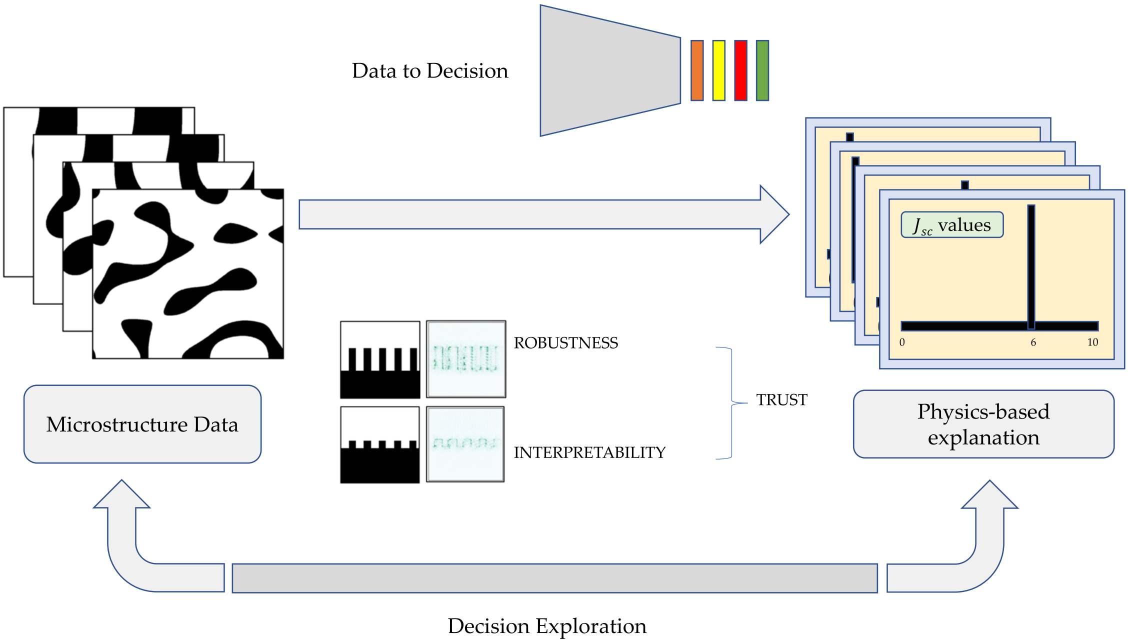


Fig. 1 DLSP (Deep Learning for Structure Property) framework: We construct a forward map from morphology to performance. Upon building trust in this trained model, we use it for performing manual exploration and insight buildings, as well as and automated design

model that is not interpretable raises a critical question—why should a domain expert believe in the prediction of a black-box model? This lack of "interpretability or explainability" is endemic to most black-box models and presents a major bottleneck to the widespread acceptance of ML models.[23] Recently, there have been several approaches towards extracting interpretation from these "black-box" models.[23–27] This includes domain-specific explanation of models.[28–31] In the current context, the process of learning the structure-property relationship involves identifying several distinct local morphological traits (i.e., unsupervised feature learning) and weighing them appropriately to predict the performance of the morphology. While several (similarly performing) architectures will learn to look at multiple features, we argue that the most useful network is the one that can also identify the right features of the morphology used to make the (correct) prediction. In other words, the chosen architecture should be interpretable to gain trust in the model.

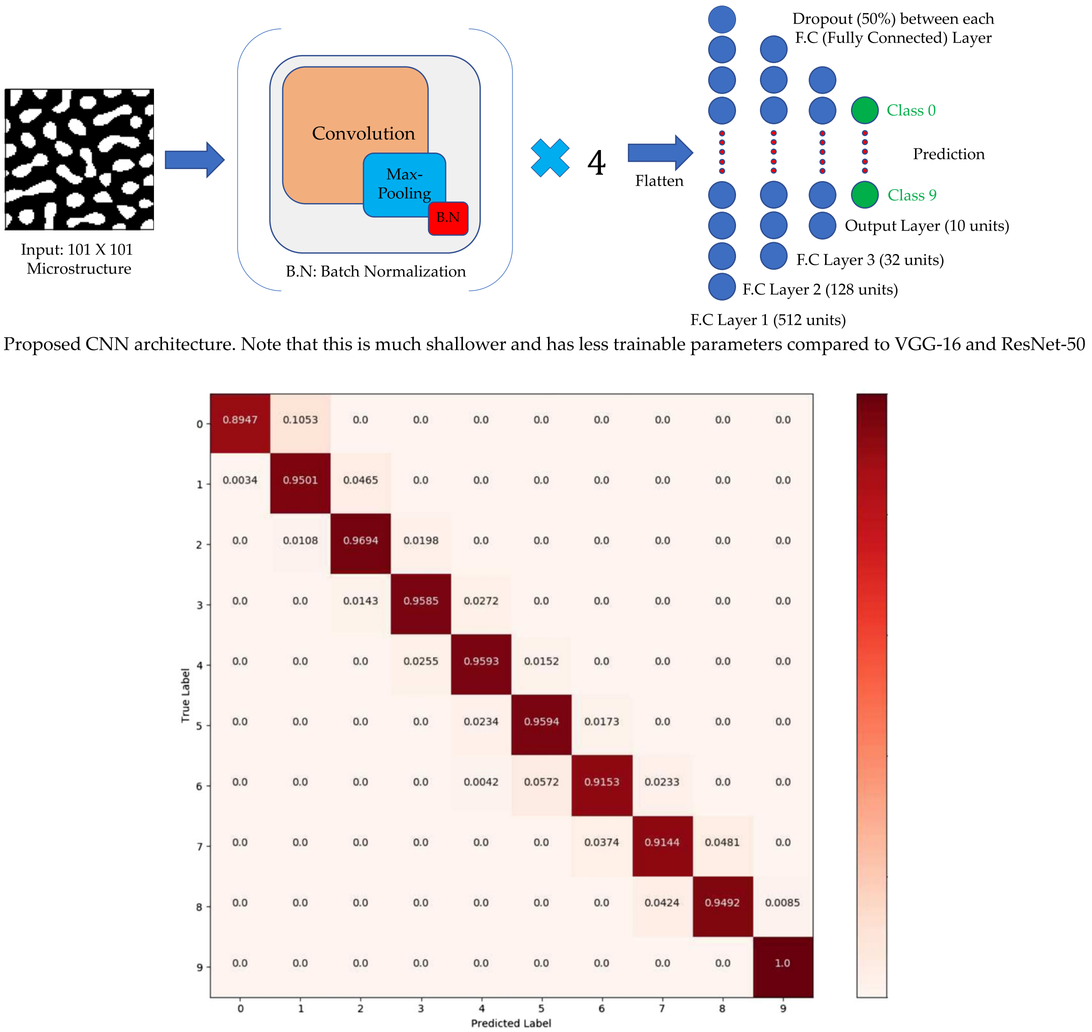
We introduce an approach called DLSP (Deep Learning for Structure Property interrogation) for learning the structure-property relationship from data. Figure 1 illustrates this approach graphically. We first construct a surrogate model of the structure-property relationship using a custom architecture based on a deep convolutional neural network. After training, this architecture is characterized for its trust using generalizability and interpretability measures. Specifically, generalizability is characterized by the performance of the models on off-sample morphologies, whose characteristics are not present in the training dataset. Subsequently, interpretability is characterized by evaluating the "salient" features using saliency map visualizations. This dual characterization allowed us to pick a custom architecture over standard classifying architectures such as VGG-16 and ResNet50 architectures, all of which had nearly identical predictive power. We further use this trust-worthy architecture to perform manual as well as automated explorations of the structure-property space. Using a graphical web application we simplified the process of manual exploration and intuition building of the structure-property space. Here, the user can manually draw (2D) microstructures, perturb the microstructures and use the trained model to rapidly explore the impact of specific features on performance. Such analysis using a full scale physics model would require established, complex computing resources, which are generally not available to every researcher. Additionally, we integrated this trained model into an optimization framework to enable automated morphology. This work illustrates the substantial promise of such surrogate based design procedures in the design of complex multi-physics systems.

RESULTS

Training and validation

We develop a CNN-based architecture to classify morphologies into performance classes. A diverse set of binary morphologies were computationally created for use in training, testing and validation. We solved a thermodynamically consistent Cahn-Hilliard equation[32] for binary phase separation using an in-house finite element library.[33] We ensured creation of a diverse set of morphologies by simulating systems with different volume fractions and different binary interaction parameters. As the Cahn-Hilliard equation models spinodal decomposition (or coarsening dynamics), we output morphologies at several time-snapshots for each simulation. A total of ~65,000 morphologies were generated. Each of these morphologies was computationally interrogated to evaluate the photovoltaic performance. The short-circuit current, J_{sc}, was evaluated for each morphology using the excitonic drift-diffusion equation,[1] which models photocurrent generation process in organic semiconducting films. Across the dataset, the J_{sc} exhibited a minimum of 0.6 mA/cm^2 and a maximum of 7.0 mA/cm^2. Subsequently, the continuous output, J_{sc} was binned into 10 distinct equi-spaced bins, and each morphology was assigned a one-hot vector as its label.

The dataset consists of images aggregated from solving the Cahn-Hilliard equations for a binary phase separating mixture with various blend ratios and interaction parameters (the complete dataset is publicly available). Varying interaction parameters produce morphologies with different domain purities, while varying blend ratios produce domains of different sizes. Here, we choose to consider 2D morphologies, with extension to 3D morphologies being conceptually straightforward (but computationally non-trivial[11,34]). This dataset of morphologies (i.e., 2D, amorphous, isotropic) chosen is a subset of the diversity of



(a) Proposed CNN architecture. Note that this is much shallower and has less trainable parameters compared to VGG-16 and ResNet-50

(b) Confusion matrix for in-sample test predictions. Notice the heavily diagonally dominant matrix, indicating a very good classification accuracy.

Fig. 2 Confusion matrix for in-sample test predictions. Notice the heavily diagonally dominant matrix, indicating a very good classification accuracy. (Scalebar limits: 0–1)

morphologies that OPV films exhibit (amorphous-crystalline, anisotropic, and multi-phase) (Interestingly, we show our model trained on this strict subset of plausible morphologies performs well on morphologies representative of the larger OPV diversity, see Section Out-of-sample testing to characterize model generalizability).

We choose the short-circuit current, J_{sc}, as the output of the model. The performance of an OPV device is characterized by the current-voltage (JV) plot. The JV plot is completely parameterized in terms of three quantities, (a) open circuit voltage V_{oc}, (b) short-circuit current J_{sc}, and (c) fill factor. The J_{sc} explicitly depends on the morphology, while V_{oc} depends on the chemistry of the acceptor-donor materials (essentially the HOMO-LUMO gap). Consequently, this motivates our choice of J_{sc} as the output since it explicitly encodes the influence of morphology. Our custom network architecture for mapping a specific morphology to its label is depicted in Fig. 2. It has 1.2 million learning parameters, consisting of four blocks comprised of a convolutional layer followed by a pooling layer (downsampling by 2 × 2 max-pooling) followed by a batch normalization layer. The first and second blocks have 16 feature maps with 5 × 5 convolutional kernels. The third block has 64 feature maps with 2 × 2 kernels and the final block has 128 feature maps with 2 × 2 kernels. After the final block, the output is flattened using a flatten layer and is followed by 3 fully connected (FC) layers with 512, 128, and 32 hidden units each, sequentially before reaching the final softmax output (prediction) layer of 10 units. A Dropout layer[35] with 50% dropout was added between each of the FC layers. Training was performed on a total of 45,108 samples (with an additional 11,109 validation samples), and testing was performed on 11,109 samples. The learning rate was initiated at 0.0001. The Rectified Linear Unit (ReLU) function is used as the activation function for each of the convolutional and dense (FC) layers. To address over-fitting issues, we add dropout layers in between the fully connected (FC) layers. The percentage of dropouts used was 50% after each of the fully connected layers (namely, FC Layer 1, FC Layer 2, and FC Layer 3, as shown in Fig. 2). After every convolutional and subsequent max-pooling layer, batch normalization was performed to remove internal covariate shift.[36] The network was trained for approximately 120 epochs (18*s* per epoch) with a learning rate of 0.0001, on the 45,000-image training set, giving an accuracy of 95.80%. The loss was denoted using a categorical cross-entropy function and Adam optimizer[37] was used to minimize the error.

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

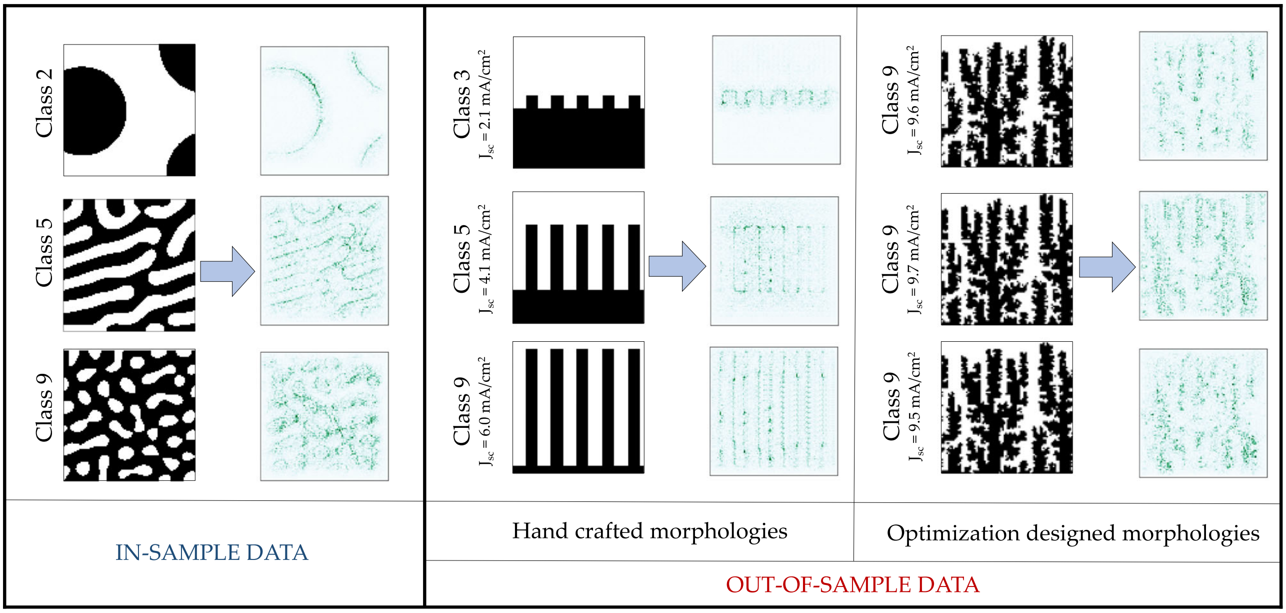


Fig. 3 Saliency maps and performance of our custom trained CNN. Note how the saliency maps closely follow the interface regions in the microstructure. It should also be noted that the networks shows good performance even on samples outside the training dataset

- 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 180*s* per epoch) with no further improvement in test accuracy.
- 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 (580*s* per epoch) with no further improvement in test accuracy.

A key point to note is that our network, although shallower, performs as well as the established deeper CNN models. Therefore, we select the network based on the learnt features ('interpretability') and out-of-sample performance ('generalizability') and not just the accuracy/f1-score of model on the testing dataset. We also note that deeper networks also have additional problems—vanishing (or exploding) gradients,[38] which hinder convergence, and the saturation of accuracy with increasing depth. We use saliency maps[27] to visualize learnt features (Sec. Building trust via interpretability characteristics), i.e., identify microstructure features used by the model to make classification decisions. It is observed that the heat-maps signify the regions of varying degrees of importance and suggest a physical interpretation, which is further discussed in Sec. Building trust via interpretability characteristics.

Performance of models: statistical metrics

A standard approach to quantify performance of a classification based machine learning framework is through the confusion matrix. Figure 2b shows the confusion matrix for in-sample test data classification. It has an accuracy of 95.80% and F1-score of 97.28%. From the confusion matrix, it can clearly be seen that most of the classification is correct, and those which are incorrectly predicted are usually only off by one class. Some incorrect prediction is not unexpected, as we are binning a continuous variable into non-overlapping classes. As such, the edge cases have the potential to be misclassified. We also note that the other two standard architectures show similar confusion matrices, with similar prediction accuracy (see SI).

Out-of-sample testing to characterize model generalizability

It is a commonly known fact[35] that neural networks can possibly overfit, depending on the model capacity, amount of training data and training hyperparameters. The network thus memorizes the data and exhibits poor generalization capacity as well as brittleness (i.e., lack of robustness to perturbations). We, therefore, 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. First, we test the network on a columnar structure (Fig. 3). This structure is postulated as an ideal structure in literature.[39] As the width of the columns decrease (and of the order of the exciton diffusion length) and the length of the columns increase, the performance of the morphology increases. This is an example of out-of-sample data—it has several sharp interface contours, which are completely absent in the training dataset. The results of the performance of the models on this morphology are shown in Fig. 3. The actual J_{sc} values from a full scale drift-diffusion simulation (along with the corresponding true label) are also presented. It is promising that the custom network accurately predicts the correct label corresponding to each of the columnar microstructures.

In a more difficult generalizability test, we use fractal-like morphologies,[40] that are constructed to maximize the interfacial area while minimizing the amount of tortuous transport. These 'virtual' morphologies have been shown to exhibit enhanced performance,[40] but are currently difficult to experimentally fabricate. We make this point to emphasize that our training dataset consists fully of morphologies that are experimentally feasible to fabricate. Our model correctly predicts the J_{sc} class of all fractal-like morphologies we considered (100% accuracy). It is very promising that our network has correctly identified (Fig. 3) all these as high-performing class label 9. This provides substantial evidence of the generalizability of the model.

Building trust via interpretability characteristics

We next query the network to characterize the learnt features. We accomplish this using the concept of saliency maps[27,41] to identify

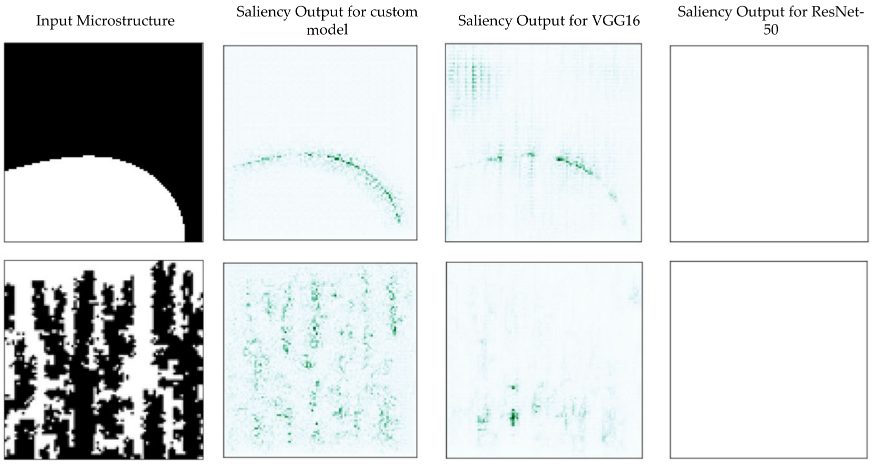


Fig. 4 Comparison of Saliency map outputs for our Custom Model (second column), VGG-net (third column) and ResNet-50 (fourth column), with input morphologies shown in the first column: top row shows an example image for class 0, bottom row shows an example image from fractal-like morphologies (correctly predicted as class 9 by our custom model)

the important features of the image input. Saliency mapping is a visualization technique that generates heat-maps on images that bring out (highlight) the regions (microstructure regions, for our case) the trained CNN model focuses on to generate a classification output. Figure 3 shows the saliency maps for morphologies in the data, columnar structures and the "high" performing morphologies identified in.[40]

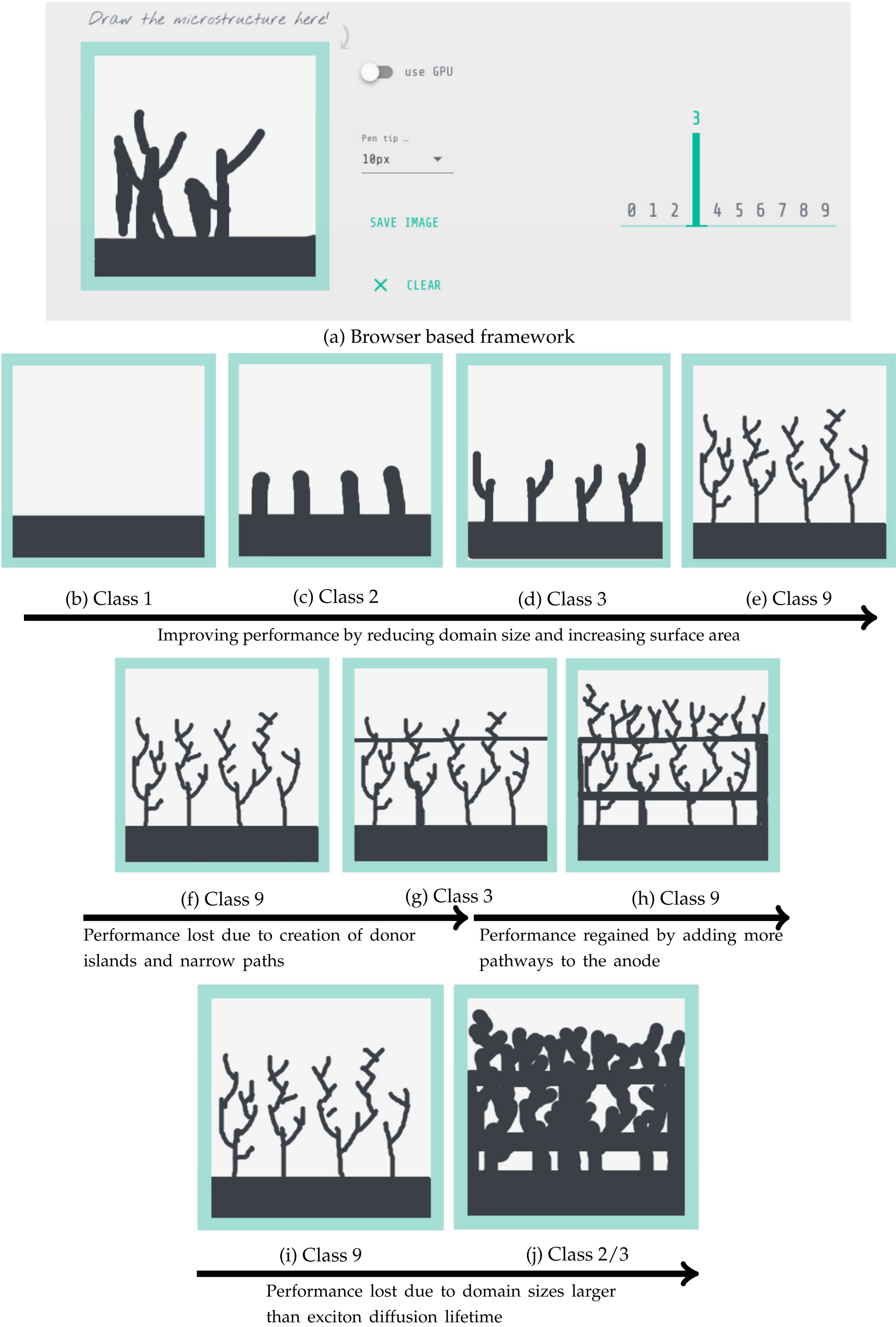
We can see, in Fig. 3, how the network uses the interface between the acceptor and donor regions feature as a key measure for prediction. We believe this is critical evidence that makes this network trust-worthy. This is because the interface is the most critical feature affecting the performance. The length of the interface determines the amount of excitons that are dissociated. Additionally, interfaces that results in isolated islands or highly tortuous pathways result in enhanced recombination thus reducing performance. Finally, the impact of interfaces in the middle of the domain (away from the top and bottom electrodes) are more important, as the charges produced at these locations have a higher chance of recombination. We can see from Fig. 3 how the network is able to identify and utilize this interface information as critical to prediction of device performance.

Finally, we observe in Fig. 4 that the saliency maps from the standard deep networks (VGG-16 and ResNet-50) are unable to locate any interpretable features. Although the test accuracy of these networks is marginally higher than our custom network, we see that the saliency outputs from these networks do not provide us with any understandable information. Extensive numerical experimentation revealed that our model is shallow enough to provide meaningful saliency maps (i.e., be intepretable) while deep enough to produce accurate (and generalizable) predictions. We provide additional details in Appendix. 4.6. This observation is in-line with,[42] 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 this customized architecture as a surrogate map from the microstructure space to the performance space.

Morphology design

Having developed a fast and trust-worthy surrogate map from microstructure to performance, we use it to enable microstructural design. In this section, we show two distinct applications, one manual and one automated, using this surrogate model for microstructures exploration and design. The goal of both these techniques is to explore and identify 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. In the OPV context, evaluating the J_{sc} for a 2D morphology requires access to dedicated high-performance computing resources. While our highly optimized in-house excitonic-drift-diffusion[1,43] code is able to perform one simulation in a few minutes on 24 processor, this is still not a viable approach for in-line design exploration and insight generation. In contrast, with the CNN-based framework, evaluating the morphology becomes significantly faster and easier. Hence it provides an very powerful way to quickly 'evolve' morphologies to reach morphologies with optimized performance.

Using the surrogate, we created a browser (Fig. 5a) that enables the user to interactively modify morphologies to both visualize, test/build intuition and improve morphology performance. Using this interface, the user can get insights into the effect of morphological features on performance. Figure 5 shows how one can modify morphologies to sequentially 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. An example of this is demonstrated in Fig. 5b–j. It shows a set of morphologies along with the respective performance labels predicted by our network. First, we can see how performance can be improved from a simple bilayer by increasing the amount of surface area between the acceptor and donor.[44] The maximum boost of performance is obtained when the donor(black) domains are fractal-like,[40] as shown in Fig. 5e. Next, we add island type structures to inhibit performance.[44] In our example, a 'line' of donor is added to the existing morphology, creating several acceptor domains unconnected to the cathode. The performance suffers drastically as informed by the physics of photoconversion.[1] This reduction can be compensated if the connectivities are improved for the acceptor, which can be seen in Fig. 5h. And finally, Fig. 5j shows how larger domains are not beneficial as they lead to geminate recombination and hence lower performance. Finally, a user can use approach as a design tool by incrementally adding 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 in real-time.



(a) Browser based framework

(b) Class 1 (c) Class 2 (d) Class 3 (e) Class 9

Improving performance by reducing domain size and increasing surface area

(f) Class 9 (g) Class 3 (h) Class 9

Performance lost due to creation of donor islands and narrow paths

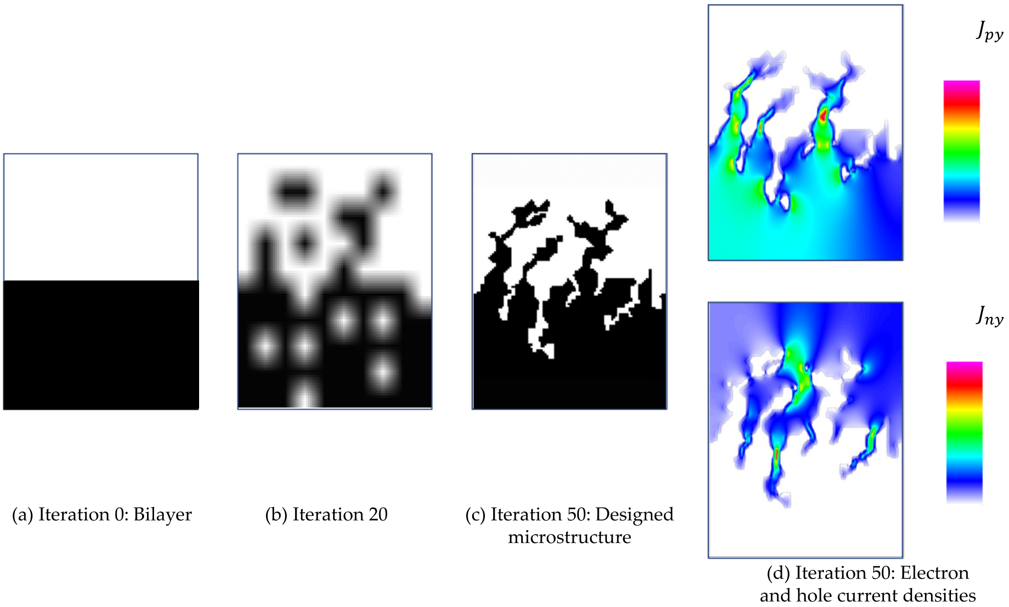
Performance regained by adding more pathways to the anode

(i) Class 9 (j) Class 2/3

Performance lost due to domain sizes larger than exciton diffusion lifetime

Fig. 5 Manual exploration and insight building using the browser interface. Notice how several physics based intuitive trends can be identified and understood by incrementally perturbing the original bilayer morphology

The above interface enables manual exploration and building of insight into the influence of various morphological features on performance. Manual exploration, however, inherently makes full exploration to find the best performing morphology manifold difficult and time-consuming. Thus, to fully explore this space, we link this fast surrogate with a probabilistic optimization algorithm to find promising, high-performing morphology classes. More specifically, we use a population based incremental learning (PBIL)



(a) Iteration 0: Bilayer (b) Iteration 20 (c) Iteration 50: Designed microstructure (d) Iteration 50: Electron and hole current densities

Fig. 6 Exploration by semi-automated design: The optimization started with a bilayer structure. Notice how the framework directs the formation of finer features. Figure **d** shows the simulated electron and hold current densities under short-circuit conditions for this optimized morphology. The result from automated design has been modified using physics based principles. (Scalebar limits: Jpy: 0–10; Jny: 0–22)

approach to perturb morphologies and evolve them towards higher performance.[40] 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 and is updated at each iteration to evolve towards promising morphology classes. 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 fast ML surrogate 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). The integration of a robust and fast surrogate with a probabilistic exploration algorithm produces very promising results. Representative results are shown in Fig. 6 where the evolution of the morphology is towards features with multiple scales, mimicking the finger-like fractal structures that are exhibited by high-performance morphologies.[40] We perform full-physics simulations on one of the optimized morphologies (Fig. 6c), which confirms that the surrogate-derived morphology is in fact a high-performing morphology (Fig. 6d).

DISCUSSION

In this work, we address the computationally challenging issue of rapidly exploring morphology space to identify promising morphologies, especially in the context of multi-physics phenomena. While the approach is general, we illustrate the approach using the case of morphology tuning to enhance the performance of organic photovoltaic films. Our approach is a data-driven approach to learn a morphology quantifier that can perform fast evaluations. We train a custom designed CNN maps a specified morphology into short-circuit current, J_{sc}, classes. Using out-of-sample datasets, we confirm absence of over-fitting issues during the training process. Two other standard networks (VGG-16 and ResNet-50) were also trained. 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. This work serves as a proof of concept of using deep neural networks for material morphology quantification and design.

There are several interesting areas of research that this work suggests. First, we show that our model—though trained on a subset of plausible morphologies—is able to make accurate predictions on a much more diverse set of morphologies. This raises the question: ‘What is the minimal diversity of morphologies that is needed for a trained model to be generalizable?’ Such questions are particularly important to answer when data collection is resource intensive. Promising approaches include methods of active learning,[45] and physics-aware models.[46,47] Next, we show that CNN-based surrogate models are promising approaches to rapidly explore structure-property manifolds. This raises the question: ‘How can such techniques be extended to map and explore process-structure-property manifolds?’ This question is particularly important to isolate promising processing windows that produce high-performing devices. Promising approaches include surrogate models based on smart sampling,[48] and ideas of manifold learning.[49]

METHODS

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 energy conversion. 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.

The solar power conversion happens in several stages. 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. These excitons dissociate at the acceptor-donor interface to form charges. The nature and quality of the interface has a direct impact on this efficiency. 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) are 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 this context, quantifying the stage efficiencies (generation, dissociation, and transport) 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[2,7] 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 deep convolutional neural networks to do performance based classification.

Data generation and quantification

In order to train the network, we generate a dataset of microstructures using a thermodynamic consistent binary phase separation simulation. This is done by solving the well known Cahn-Hilliard equation,[32] which tracks the local volume fraction of each material (ϕ_i):

$$\begin{aligned}\frac{\partial\phi_i}{\partial t} &= \nabla(M(\phi_i)\nabla\mu_i)\\ \mu_i &= \frac{\partial f}{\partial\phi_i} - \epsilon^2\nabla^2\phi_i\end{aligned} \tag{1}$$

$M(\phi_i)$ is the mobility of component i. μ_i represents the chemical potential of component i. The chemical potential as defined in Eq. (1) is the variational derivative of the total free energy of the system. The total free energy comprises of the bulk free energy f and the interfacial energy. The interfacial free energy is characterized as $0.5\epsilon^2|\nabla\phi_i|^2$, where ϵ is the interfacial energy parameter. ϵ is usually correlated with the thickness of the interface between the components. The bulk free energy is described using the Flory-Huggins[50] energy representation:

$$f = \frac{\phi_1}{N_1}\ln\phi_1 + \frac{\phi_2}{N_2}\ln\phi_2 + \chi_{12}\phi_1\phi_2 \tag{2}$$

The degree of polymerization of the components is denoted by N_i and χ_{ij} represents the severity of interaction between the components. The values for χ are either estimated using molecular simulations[51,52], or experimentally[53], or calculated through empirical methods[54].

This process generates time series of morphologies that can be treated as independent morphologies for the sake of training a machine learning model. This method helps to quickly produce several thousands of microstructures within a very short amount of time. In order to generate numerous consistent morphologies, we perform 100 simulations of the above Eq. (10) values of χ_{12} with 10 values of initial concentration), with morphologies outputted at every 20 timesteps (which provides distinguishable morphologies across timesteps). Previous analysis using this data can be found in.[44] A characteristic of this procedure for generating morphologies through simulation is their similarity to morphologies in real active layers produced during thermal annealing, for example, 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 creating morphologies that break these characteristics and testing the performance of the trained network on such samples. We produce a dataset of nearly 65,000 (2D) gray-scale morphologies of size 101 × 101pix.

These morphologies were then characterized using an in-house physics based simulator.[1] This simulator uses steady state excitonic drift-diffusion equation to model the processes of exciton dissociation and charge transport:

$$\boldsymbol{J_n} = -qn\mu_n\nabla\varphi + qV_t\mu_n\nabla n \tag{3}$$

$$\boldsymbol{J_p} = -qp\mu_p\nabla\varphi - qV_t\mu_p\nabla p \tag{4}$$

$$\nabla.\boldsymbol{J_n} = qfR_{[n,p]} - qfD_{[\nabla\varphi,X]} \tag{5}$$

$$-\nabla.\boldsymbol{J_p} = qfR_{[n,p]} - qfD_{[\nabla\varphi,X]} \tag{6}$$

$$\nabla.(\epsilon_r\epsilon_0\nabla\varphi) = q(n-p) \tag{7}$$

$$-\nabla.(V_t\mu_x\nabla X) - fD_{[\nabla\varphi,X]} - R_{[x]} = -G - R_{[n,p]} \tag{8}$$

where μ_n, μ_p are the mobilities of electrons and holes, respectively. The quantities of interest are the electrostatic potential in the active layer φ, electron density n, hole density p and exciton density X. G, $D_{[\bigtriangledown\phi,X]}$ represent the rate of generation and dissociation of excitons, respectively. $R_{[x]}$ is the exciton relaxation rate. J_n, J_p are the current densities of electrons and holes, respectively. We use the short-circuit current J_{sc} as a means of labelling the data. The whole data were divided into 10 classes, which are equally spaced between the best ($J_{sc} = 7\ mA/cm^2$) and worst performing ($J_{sc} = 0.2\ mA/cm^2$) in the data.

Convolutional Neural Networks (CNNs)

CNNs have become the standard frameworks when it comes to computer vision tasks in recent times. To serve our purpose of classifying microstructures, we also use a CNN-based model to train on our dataset, establish trust in the trained model and then use that trained model to make test/future predictions.

CNNs achieve a high level of performance with fewer parameters to learn[55,56] when compared to networks constructed simply via Fully-Connected (FC) layers. By design, they exploit the two-dimensional (2D) structure of an input image by preserving the locality of features and utilize spatially local correlations of an image by using tied weights, which are invariant to the translation of the feature positions.[55,57]

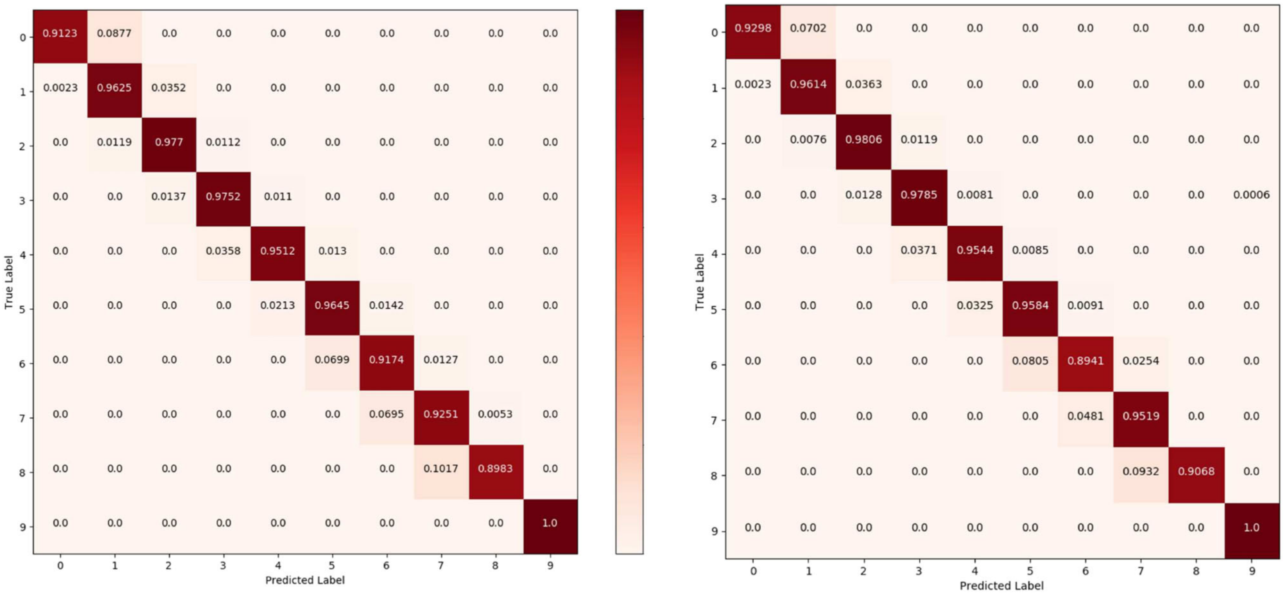
In CNNs, data are represented by multiple feature maps in each hidden layer. These feature maps are obtained by performing a local convolution of the input image using multiple filters. These feature maps further undergo non-linear downsampling with a max-pooling operation[58] to decrease the data-dimension. Max-pooling partitions the input image into sets of non-overlapping rectangles and uses the maximum value for each partition as the output. This is done so that neighboring pixels in an image sharing similar features can be discarded. Both spatial and feature abstractness are also increased as a result, imparting increased position invariance for the filters.[58,59]

We use batch normalization layers, which normalize the activations of the previous layer at each batch, to improving the overall performance of the architecture. Batch Normalization applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.[36]

Post max-pooling, multiple dimension-reduced vector representations of the input are acquired, and the process is repeated in the next layer to achieve a higher-level representation of the data. At the final pooling layer, the resultant outputs are linked to the FC layer, where Rectified Linear Unit (*ReLU*) activation outputs[60] from the hidden units are joined to output units to infer a predicted class on the basis of the highest joint probability given the input data. Keeping this in mind, the probability of an input vector v being a member of the class i can be written as follows:

$$Pr(Y = i|\mathbf{v}, \mathbf{W}, \mathbf{b}) = \text{softmax}_i(\mathbf{Wv} + \mathbf{b}) = \frac{e^{W_i v + b_i}}{\sum_j e^{W_j v + b_j}} \tag{9}$$

where elements of **W** denote the weights and elements of **b** denote the



(a) Confusion matrix for ResNet-50

(b) Confusion matrix for VGG-16

Fig. 7 Both the standard architectures show performance similar to our custom architecture. But these do not provide any meaningful explanations to their predictions (Fig. 3) (Scalebar limits: 0–1)

Table 1. Effect of depth of network on generalizability

Model	Accuracy of in-sample morphologies	Accuracy of fractal-like morphologies	Accuracy of columnar morphologies (loss value)
Model α	96%	100%	90% (4.22)
Model α_{s1}	95%	96%	90% (14.56)
Model α_{s2}	95%	57%	90% (18.34)

biases. The model prediction is the class with the highest probability:

$$y_{\text{pred}} = \text{argmax}_i Pr(Y = i|\mathbf{v}, \mathbf{W}, \mathbf{b}) \tag{10}$$

The model weights, **W**, and biases, **b**, are optimized using error back-propagation algorithm,[61] wherein true class labels are compared against the model prediction by using an error metric/loss function. We choose categorical cross entropy[62] as the loss function, chosen to be minimized for the dataset **V**, and is given as follows:

$$\mathcal{L}(V, Y) = -\frac{1}{n}\sum_{i=1}^{n} \mathbf{y}^{(i)} \ln \mathbf{a}(\mathbf{v}^{(i)}) + (1 - \mathbf{y}^{(i)}) \ln (1 - \mathbf{a}(\mathbf{v}^{(i)})) \tag{11}$$

Here, $\mathbf{V} = \{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ is the set of input examples in the training dataset, and $\mathbf{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(n)}\}$ is the corresponding set of labels for those input examples; $\mathbf{a}(\mathbf{v})$ represents the output of the neural network given an input **v**.

Class specific visualization: Saliency Maps

A detailed description of Saliency maps and their use in visualising class specific regions as learnt by CNNs has been given in ref. [27] However, we here give a brief overview as well for the sake of simplicity. Saliency Map generation is a technique, which takes an input image, a learnt classification CNN model and a class of interest as it's input and generates as an output, an image that is representative of that particular class in terms of what that learnt CNN model sees in the given input image. Formally, we define this as follows: Say, $a_i(A)$ is the score of class i, computed by the classification layer of the CNN for an image A. The target is to find a L_2-regularized image such that $a_i(A)$ is high:

$$\underset{A}{\text{argmax}}\, a_i(A) - \gamma \|A\|_2^2 \tag{12}$$

where γ is the regularization parameter. Using the back-propagation algorithm (which is also used to optimize the layer weights), we obtain a locally optimal A by optimizing with respect to the input image, with the model weights fixed to those obtained at the best-training step.

Performance of standard architectures

As discussed in Sec. Training and validation, we tested the performance of our custom architecture with standard cpnvolutional network architectures, namely ResNet-50[63] and VGG-16.[12] ResNet50 is a 50 layer deep convolutional network pretrained on images from ImageNet and can classify into 1000 object categories. It uses a special architecture called residual network blocks that simultaneously reduce the model size and capture diversity of input images. The final layer was modified to classify into 10 categories and was trained end-to-end with our data. VGG16 is another very popular architecture tested on data from ImageNet, which uses 13 layers of 3 × 3 convolutions with max-pooling followed by two fully connected layers of 4096 neurons each. As with ResNet50, we modify the final layer of VGG16 to classify into only 10 categories. Although our architecture is shallower, it showed similar performance in terms of the confusion matrix. The confusion matrix on validation data for ResNet-50 and VGG-16 are in Fig. 7.

How shallow can the network be?

In order to determine the simplest model with desired generalizable and interpretable characteristics, we performed an analysis of shallower variants of the presented architecture (Model α). We trained a shallower model (Model α_{s1}) retaining the first 3 convolution-max-pool-BN blocks of Model α (i.e., removing the last block from Model α) as well an even shallower model, α_{s2} which retains first two blocks of Model α (i.e., we remove the last 2 blocks from Model α). Table 1 compiles the performance results of these models on three test datasets: in-sample morphologies, fractal-like morphologies, and columnar morphologies. We observe that progressively shallower models perform worse in terms of prediction accuracy, especially for the out-of-sample data (fractal-like and columnar morphologies). In other words, generalizability suffers when the models become shallower than the presented model (Model α). This evidence suggests that Model α is the shallowest model that still produces viable accuracy.

The accuracy values, especially in the case of columnar morphologies is slightly misleading because it considers all wrong classifications as equally bad, irrespective of how close is the prediction to the original class. Hence, we analyzed the weighted categorical cross entropy loss for the columnar morphologies, included in paranthesis in the above table.

DATA AVAILABILITY

The dataset and the trained model used to generate the results are available through a Google Form request accessible through GitHub: https://github.com/vizer1993/GuidedStructurePropertyExploration.

CODE AVAILABILITY

The code used for the above analysis is openly available at GitHub: https://github.com/vizer1993/Photovoltaics_CNN_Surrogate

Received: 10 April 2019; Accepted: 30 August 2019;
Published online: 01 October 2019

REFERENCES

1. Kodali, H. K. & Ganapathysubramanian, B. Computer simulation of heterogeneous polymer photovoltaic devices. *Model. Simul. Mater. Sci. Eng.* **20**, 035015 (2012).
2. Wodo, O., Tirthapura, S., Chaudhary, S. & Ganapathysubramanian, B. A graph-based formulation for computational characterization of bulk heterojunction morphology. *Org. Electron.* **13**, 1105–1113 (2012).
3. Casalegno, M., Raos, G. & Po, R. Methodological assessment of kinetic monte carlo simulations of organic photovoltaic devices: the treatment of electrostatic interactions. *J. Chem. Phys.* **132**, 094705 (2010).
4. Meng, L. et al. Dynamic monte carlo simulation for highly efficient polymer blend photovoltaics. *J. Phys. Chem. B* **114**, 36–41 (2009).
5. Marsh, R., Groves, C. & Greenham, N. C. A microscopic model for the behavior of nanostructured organic photovoltaic devices. *J. Appl. Phys.* **101**, 083509 (2007).
6. Watkins, P. K., Walker, A. B. & Verschoor, G. L. Dynamical monte carlo modelling of organic solar cells: the dependence of internal quantum efficiency on morphology. *Nano Lett.* **5**, 1814–1818 (2005).
7. Marsh, R. A., Hodgkiss, J. M. & Friend, R. H. Direct measurement of electric field-assisted charge separation in polymer: fullerene photovoltaic diodes. *Adv. Mater.* **22**, 3672–3676 (2010).
8. Hwang, I.-W., Moses, D. & Heeger, A. J. Photoinduced carrier generation in p3ht/pcbm bulk heterojunction materials. *J. Phys. Chem. C.* **112**, 4350–4354 (2008).
9. Hoppe, H. & Sariciftci, N. S. Organic solar cells: an overview. *J. Mater. Res.* **19**, 1924–1945 (2004).
10. Giridharagopal, R., Shao, G., Groves, C. & Ginger, D. S. New spm techniques for analyzing opv materials. *Mater. Today* **13**, 50–56 (2010).
11. Nagasubramanian, K. et al. Explaining hyperspectral imaging based plant disease identification: 3d cnn and saliency maps. arXiv preprint arXiv:1804.08831 (2018).
12. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).
13. Arad, B., Ben-Shahar, O. & Timofte, R. Ntire 2018 challenge on spectral reconstruction from rgb images. in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, 929–938 (2018).
14. Graves, A., Mohamed, A.-r. & Hinton, G. Speech recognition with deep recurrent neural networks. in *2013 IEEE Int. Conf. on Acoustics, Speech And Signal Processing*, 6645–6649 (IEEE, 2013).
15. Wang, T., Wu, D. J., Coates, A. & Ng, A. Y. End-to-end text recognition with convolutional neural networks. in *Proc. 21st Int. Conf. on Pattern Recognition (ICPR2012)*, 3304–3308 (IEEE, 2012).
16. Covington, P., Adams, J. & Sargin, E. Deep neural networks for youtube recommendations. in *Proc. 10th ACM Conf. on Recommender Systems*, 191–198 (ACM, 2016).
17. Bojarski, M. et al. End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016).
18. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. & Stefanovic, D. Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intell. Neurosci.* **2016**, 11 (2016).
19. Stoecklein, D., Lore, K. G., Davies, M., Sarkar, S. & Ganapathysubramanian, B. Deep learning for flow sculpting: insights into efficient learning using scientific simulation data. *Sci. Rep.* **7**, 46368 (2017).
20. Ghadai, S., Balu, A., Sarkar, S. & Krishnamurthy, A. Learning localized features in 3d cad models for manufacturability analysis of drilled holes. *Computer Aided Geometric Des.* **62**, 263–275 (2018).
21. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: generative models for matter engineering. *Science* **361**, 360–365 (2018).
22. Dieb, T. M. & Tsuda, K. *Nanoinformatics* pp. 65–74 (Springer, Singapore, 2018).
23. Castelvecchi, D. Can we open the black box of ai? *Nat. News* **538**, 20 (2016).
24. Selvaraju, R. R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. https://arxiv.org/abs/1610.02391v3 (2016).
25. Ribeiro, M. T., Singh, S. & Guestrin, C. "why should i trust you?": Explaining the predictions of any classifier. in *Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144 (ACM, New York, 2016). https://doi.org/10.1145/2939672.2939778.
26. Shrikumar, A., Greenside, P. & Kundaje, A. Learning important features through propagating activation differences. in *Proc. 34th Int. Conf. On Machine Learning (ICML-17)* (2017).
27. Simonyan, K., Vedaldi, A. & Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint 1312.6034 (2013).
28. Ghosal, S. et al. An explainable deep machine vision framework for plant stress phenotyping. *Proc. Natl Acad. Sci.* **115**, 4613–4618 (2018).
29. Holzinger, A., Biemann, C., Pattichis, C. S. & Kell, D. B. What do we need to build explainable ai systems for the medical domain? arXiv preprint arXiv:1712.09923 (2017).
30. Toda, Y. et al. How convolutional neural networks diagnose plant disease. *Plant Phenomics* **2019**, 9237136 (2019).
31. Lee, H. et al. An explainable deep-learning algorithm for the detection of acute intracranial haemorrhage from small datasets. *Nat. Biomed. Eng.* **3**, 173 (2019).
32. Cahn, J. W. On spinodal decomposition. *Acta Metall.* **9**, 795–801 (1961).
33. Wodo, O. & Ganapathysubramanian, B. Modeling morphology evolution during solvent-based fabrication of organic solar cells. *Computational Mater. Sci.* **55**, 113–126 (2012).
34. Ghadai, S., Balu, A., Krishnamurthy, A. & Sarkar, S. Learning and visualizing localized geometric features using 3d-cnn: An application to manufacturability analysis of drilled holes. In *Interpretability Symposium at the 31st Neural Information Processing Systems (NIPS-17)* (2017).
35. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
36. Ioffe, S. & Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. in *Proc. Int. Conf. on Machine Learning* 448–456 (2015).
37. Kingma, D. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint 1412.6980 (2014).
38. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *AISTATS* **9**, 249–256 (2010). http://proceedings.mlr.press/v9/glorot10a.html.
39. Kodali, H. K. & Ganapathysubramanian, B. Sensitivity analysis of current generation in organic solar cells—comparing bilayer, sawtooth, and bulk heterojunction morphologies. *Sol. energy Mater. Sol. cells* **111**, 66–73 (2013).
40. Du, P., Zebrowski, A., Zola, J., Ganapathysubramanian, B. & Wodo, O. Microstructure design using graphs. *npj Computational Mater.* **4**, 50 (2018).
41. Montavon, G., Samek, W. & Müller, K. Methods for interpreting and understanding deep neural networks. *Digital Signal Proc.* **73**, 1–15 (2017). https://doi.org/10.1016/j.dsp.2017.10.011.
42. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. & Lipson, A. Understanding neural networks through deep visualization. arXiv preprint arXiv: 1506.06579 (2015).
43. Kodali, H. K. & Ganapathysubramanian, B. A computational framework to investigate charge transport in heterogeneous organic photovoltaic devices. *Computer Methods Appl. Mech. Eng.* **247–248**, 113–129 (2012).
44. Wodo, O., Zola, J., Pokuri, B. S. S., Du, P. & Ganapathysubramanian, B. Automated, high throughput exploration of process–structure–property relationships using the mapreduce paradigm. *Mater. Discov.* **1**, 21–28 (2015).
45. Pokuri, B. S. S., Lofquist, A., Risko, C. M. & Ganapathysubramanian, B. Paryopt: A software for parallel asynchronous remote bayesian optimization. arXiv preprint arXiv:1809.04668 (2018).
46. Singh, R. et al. Physics-aware deep generative models for creating synthetic microstructures. arXiv preprint arXiv:1811.09669 (2018).
47. Shah, V. et al. Encoding invariances in deep generative models. arXiv preprint arXiv:1906.01626 (2019).
48. Pfeifer, S., Pokuri, B. S. S., Du, P. & Ganapathysubramanian, B. Process optimization for microstructure-dependent properties in thin film organic electronics. *Mater. Discov.* **11**, 6–13 (2018).

49. Schoeneman, F., Chandola, V., Napp, N., Wodo, O. & Zola, J. Entropy-isomap: Manifold learning for high-dimensional dynamic processes. in *2018 IEEE Int. Conf. on Big Data (Big Data)*, 1655–1660 (IEEE, 2018).
50. Flory, P. J. Thermodynamics of high polymer solutions. *J. Chem. Phys.* **10**, 51–61 (1942).
51. Fu, Y.-T., Risko, C. & Bredas, J.-L. Intermixing at the pentacene-fullerene bilayer interface: a molecular dynamics study. *Adv. Mater.* **25**, 878–882 (2013).
52. Kok, C. M. & Rudin, A. Prediction of flory–huggins interaction parameters from intrinsic viscosities. *J. Appl. Polym. Sci.* **27**, 353–362 (1982).
53. Orwoll, R. A. The polymer-solvent interaction parameter x. *Rubber Chem. Technol.* **50**, 451–479 (1977).
54. Hansen, C. M. *Hansen Solubility Parameters, A User's Handbook*, 2nd edn (CRC Press, 2007).
55. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105 (2012).
56. LeCun, Y. & Bengio, Y. *The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series.* MIT press. 255–258 (1998).
57. Lee, H., Grosse, R., Ranganath, R. & Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, 609–616 (ACM, 2009).
58. Boureau, Y.-L., Bach, F., LeCun, Y. & Ponce, J. Learning mid-level features for recognition. in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2559–2566 (IEEE, 2010).
59. Huang, F. J., et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. in *Computer Vision and Pattern Recognition*, 2007. CVPR'07. IEEE Conference on, 1–8 (IEEE, 2007).
60. Nair, V. & Hinton, G. E. Rectified linear units improve restricted boltzmann machines. in *Proc. 27th Int. Conf. On Machine Learning (ICML-10)*, 807–814 (2010).
61. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
62. Rubinstein, R. The cross-entropy method for combinatorial and continuous optimization. *Methodol. Comput. Appl. Probab.* **1**, 127–190 (1999).
63. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. eprint. arXiv preprint arXiv:0706.1234 (2015).

ACKNOWLEDGEMENTS

S.G., A.K., and S.S. were funded by AFOSR YIP FA9550-17-1-0220 and DARPA HR00111990031. B.S.S.P. and B.G. were funded by NSF DMREF 1435587 and DARPA HR00111990031. XSEDE computational resources were used for microstructure simulation and quantification. We gratefully acknowledge financial support from all the above agencies.

AUTHOR CONTRIBUTIONS

B.S.S.P. contributed to project oversight, data generation and curation, architecture design and analysis, and manuscript preparation and revision. S.G. contributed to architectural refinement and comparison with standard models, saliency map visualization and manuscript review. A.K. contributed to saliency map visualization. S.S. and B.G. contributed to problem formulation, project oversight and manuscript preparation and revision.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to S.S. or B.G.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.