PAPER

Check for updates

Distinguishing liquid crystalline nematic variants by machine learning

Alexander R. Quinn,[a] Rebecca Walker, [iD][b] Naila Tufaha,[b] John MD Storey,[b] Corrie T. Imrie [iD] †[b] and Ingo Dierking [iD] *[a]

Two different machine learning architectures – sequential convolutional neural networks (CNN) and parallel inception models were evaluated with respect to their ability to identify nematic liquid crystal variants, including the ferroelectric and the twist-bend nematic phases. Varying levels of model complexity were employed from 1- to 5-layer CNNs, to 1- to 3-block inception models. Various types of augmentations like flip, contrast and brightness were used, together with dropout-layer regularisation. Flip was the only augmentation trialled to yield positive results with an acceptable level of accuracy and error, while the inclusion of dropout regularisation almost exclusively led to lower accuracies. From the systematic investigation it is advised that different variants of the nematic phase can be distinguished to an accuracy better than $0.96-0.98 \pm 0.01$ by the use of 3-layer CNNs or a model with a single inception block, if flip augmentation is applied. Computational restraints therefore suggest that a sequential CNN is sufficient to characterise phase sequences with four or fewer different phases. Higher accuracies, closer to 100%, can be achieved for extended and class-balanced datasets. In the latter case an inception approach would possibly be beneficial, depending on the size of the dataset, but overfitting needs to be avoided.

Introduction

For more than a century, liquid crystal (LC) phases have been characterised by polarising optical microscopy (POM), exploiting the wonderful and colourful world of textures, structures and defects provided by the optics of anisotropic fluids.[1–4] Nevertheless, this method is still largely based on practice and experience, because POM can only provide indications of transition temperatures and leaves the actual phase characterisation to the qualified guess of the researcher investigating the liquid crystal textures. For a more detailed characterisation, other methods need to be employed in addition, such as differential scanning calorimetry (DSC),[5] which provides phase transition temperatures and the order of respective transitions, but no indication of the actual LC phase. The actual phase structure can only be obtained by X-ray diffraction,[6] which is time consuming and, in many cases, not experimentally trivial.

In recent years, a fourth method of phase characterisation has been established in the form of machine learning *via* convolutional neural networks (CNN) and other algorithms.[7]

Naturally, this started with the distinction between the isotropic and the nematic phase, thus the simple case of dark *vs.* bright,[8–11] which is used for example in the automatic readout of liquid crystal sensors.[12] Work was mainly carried out on thermotropic nematics with their characteristic schlieren texture; while the training of algorithms was performed mostly with simulated textures,[13] some experimental studies[14] have been reported. It was not until recently,[15] that the characterisation of liquid crystal phases was expanded to various other phases with algorithm training being performed on experimentally obtained textures. It was demonstrated that nematic, fluid smectic, hexatic smectic, and soft crystal phases can be distinguished and characterised with good accuracies of approximately 95%[15]; and even continuous second order transitions like SmA–SmC were surprisingly easy to be distinguished.[16] Further successful experiments were carried out on transitions involving the soft crystal B phase[17] and glasses,[18] while other transitions like the fluid SmA to hexatic SmB phase are still somewhat illusive,[19] due to the absence of any distinguishing features in the textures of both phases. Chiral phases, like the fluid sub-phases exhibiting paraelectric, ferroelectric, ferri- and antiferroelectric behaviour, could also be well distinguished and characterised.[20]

Despite all the success in the application of machine learning algorithms to liquid crystals in the last few years, it is also of importance to realise the limitations of this approach. A good

[a] *Department of Physics and Astronomy, University of Manchester, Oxford Road, Manchester, M13 9PL, UK. E-mail: ingo.dierking@manchester.ac.uk*

[b] *Department of Chemistry, School of Natural and Computing Sciences, University of Aberdeen, Meston Walk, Aberdeen, AB24 3UE, UK*
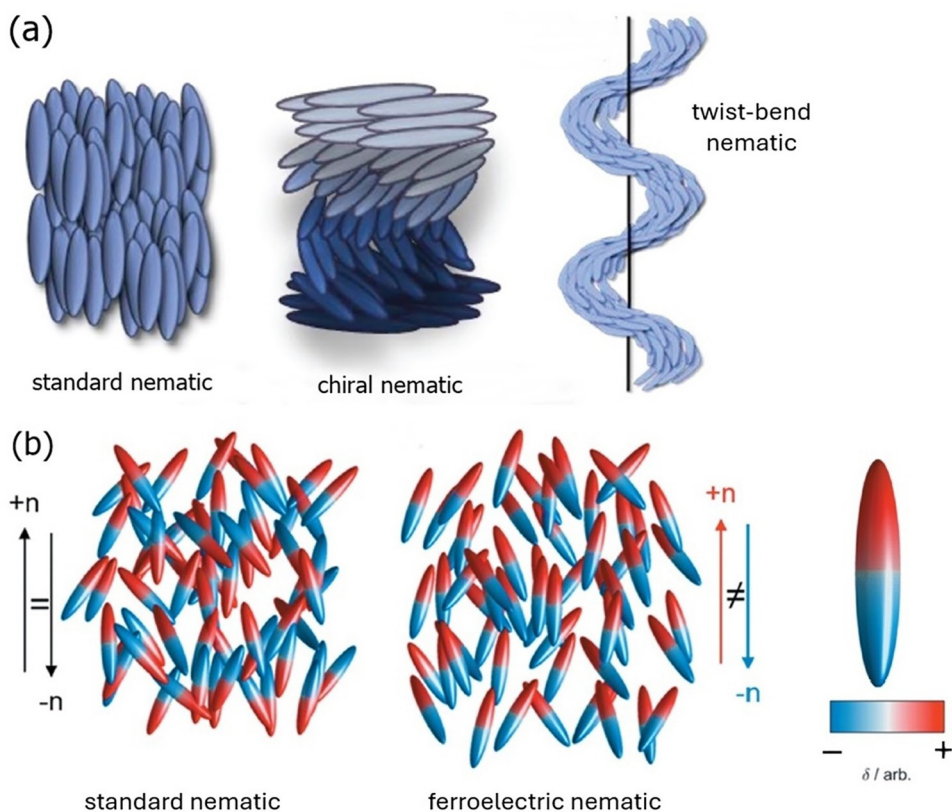
† Deceased.

Fig. 1 (a) Schematic illustration of the standard nematic phase with orientational order, its chiral counterpart, the cholesteric phase which exhibits a macroscopic helical superstructure, and the twist–bend nematic phase which locally spirals around a preferred direction. (figure reproduced by permission after[38]). (b) In the ferroelectric nematic phase, the head–tail symmetry $\mathbf{n} = -\mathbf{n}$ of the standard nematic phase is broken, and the molecular electric dipoles align approximately parallel, leading to the formation of a spontaneous polarisation whose direction can be reversed between two stable states by reversal of an applied electric field. (figure reproduced by permission after[39]).

quality set of training data is of utmost importance to achieve decent results, which obviously implies the correct labelling of phases. One further criterium was already mentioned above: different phases need to exhibit different varying features, which is not always the case, as for example in the transition between SmA to hexatic SmB, where sometimes no differences in textures are observed when the transition is passed.[19] Another point of importance is that the individual textures for a particular phase need to show some variation, otherwise, the algorithm will show pronounced overfitting. A similar effect is observed for datasets that are too small. In our experience it is best to have at least 1000 images per phase, unless the phase is completely different from the others, like the isotropic or the crystalline phase, for which fewer images may be sufficient. For example, in the simple yes–no classification problem of a LC sensor, fewer images are permittable. Further, it is important to rely on a balanced dataset of approximately equal numbers of images for each liquid crystal phase, otherwise, the analysis will be biased towards the phase with the larger number of images.[15] At last, the complexity of the machine learning algorithm employed should be matched to that of the problem to be investigated, in the case of an over-complex model, overfitting and a reduced accuracy is observed.[21] A detailed investigation of the factors influencing the performance of CNNs can be found in ref. 22.

The nematic is probably the best studied and most well-known of the liquid crystal phases, due to its broad range of applications. It is the least ordered of the liquid crystal phases, and the one with the highest symmetry. Until some time ago it was thought that the thermotropic nematic phase exhibits a structure with only uniaxial orientational order of the long axis of calamitic (rod-like) molecules. The first observation of a biaxial nematic was then suggested[23,24]; a much-discussed question which does not seem to have been resolved to date.[25] A nematic variant which has indeed been confirmed beyond doubt is the twist-bend nematic (N_{TB}) phase,[26,27] which has recently been reviewed,[28] also with respect to chemistry,[29] theory,[30] and applications.[31] A more recent variant is the long sought after ferroelectric nematic phase (N_F),[32,33] with a very informative summary provided in,[34] and reviews published with respect to chemistry,[35] theory,[36] as well as properties and applications.[37]

Both the twist-bend nematic and the ferroelectric nematic phases have been schematically illustrated in Fig. 1(a) and (b), respectively, in comparison to the standard thermotropic nematic phase composed of calamitic molecules. The standard nematic phase exhibits orientational order of the long axis of rod-like molecules along an average direction called the director $\mathbf{n}$, while the centres of mass are isotropically distributed.
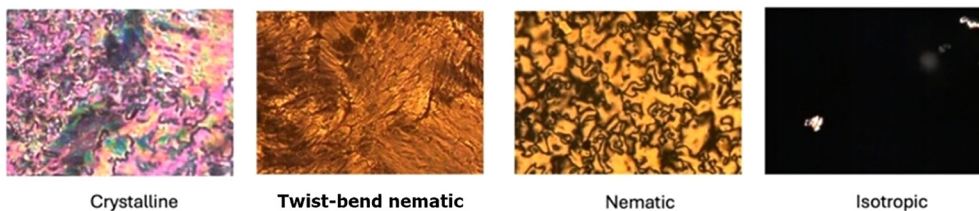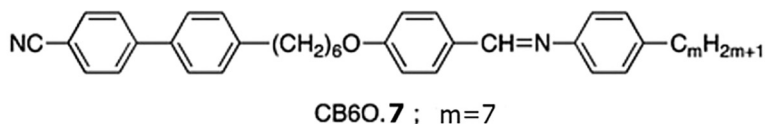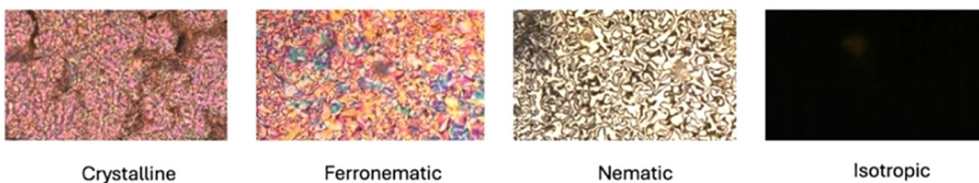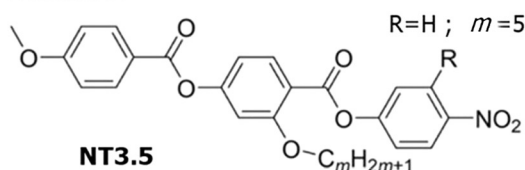
Fig. 2 Structural formulae and representative textures of the materials investigated, (a) the twist-bend nematic **CB6O.7**[40] and (b) the ferroelectric nematic **NT3.5**.[41] The longer length of the texture images corresponds to 860 μm.

The director is a pseudo-vector which shows head–tail symmetry, thus $\mathbf{n} = -\mathbf{n}$. For reasons of completeness, we should mention that the nematic phase of chiral molecules (chiral nematic, cholesteric phase) exhibits a helical superstructure with a pitch of the order between 100 nm to many μm. In the twist-bend nematic phase the molecules spiral around a preferred direction with a pitch which is extremely small, consisting of approximately 10 molecules.

For the ferroelectric nematic phase, the common head–tail symmetry of $\mathbf{n} = -\mathbf{n}$ is broken and the molecular electric dipole moments do not compensate across small spatial dimensions. The structure therefore exhibits a spontaneous polarisation which can be switched between two polar states by reversal of an applied electric field.

In this study we demonstrate that the different nematic variants, as well as the isotropic and the crystalline phase can be distinguished by machine learning via convolutional neural networks and inception models.

Experimental

Materials, image acquisition and experimental input

The twist-bend nematic material investigated in this study is a homologue of a series of 1-(4-cyanobiphenyl-4′-yl)-6-(4-alkyl-anilinebenzylidene-4′-oxy)hexanes abbreviated as **CB6O.7** and

reported by Walker et al. in ref. 40. The molecular structure is depicted in Fig. 2(a), together with a selection of characteristic textures observed. The N_{TB} phase is monotropic and the phase sequence is given by Cr. 89 N_{TB} (73) N 109 Iso. (temperatures in °C).

The molecular structure of the ferroelectric nematic phase is provided in Fig. 2(b), together with respective textures. The compound was reported by Tufaha et al. in ref. 41. Its nematic and ferroelectric nematic phases are also monotropic, with the phase sequence given by Cr. 102 N_F (63) N (68) Iso. (temperatures in °C). We note that **CB6O.7** exhibits a thread-like texture of the standard nematic phase, while **NT3.5** shows that of a schlieren texture with topological defects, besides the N_{TB} and N_F textures.

The texture images to create a dataset were frame grabbed from a number of different movies taken at different positions of the sample between untreated glass plates with optical polarising microscopy (POM, Leica DMLP). This was equipped with a Linkam LTSE350 hot stage and a TP94 temperature controller for relative temperature accuracies of ±0.1 K. Movies were recorded on cooling, at rates between 0.1–0.5 K min⁻¹ at 10 frames per second (fps) with a IDS uEye digital camera. Care was taken to generate images that were different from each other to prevent the employed machine learning algorithms to learn textures "by heart". Images of 2048 × 1088 pixel resolution were extracted using the video scene filter in the VLC media player.[42] Depending on the rapidity of changes in
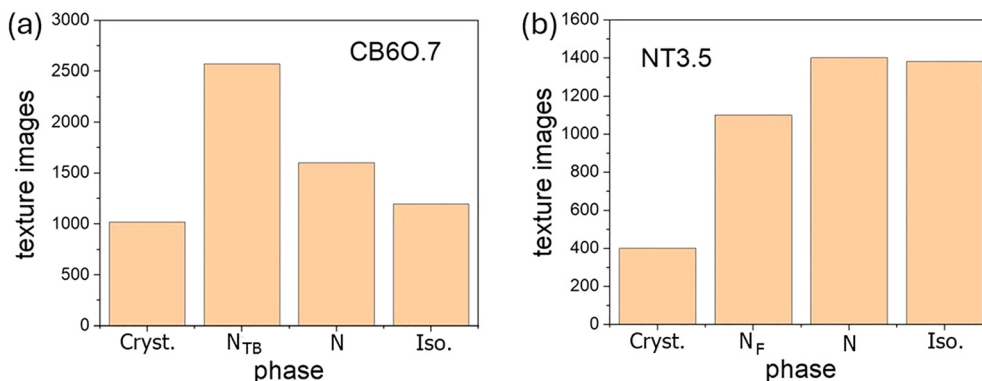
Fig. 3 Number of images generated for the different phases of (a) the twist-bend nematic **CB6O.7** and (b) the ferroelectric nematic **NT3.5**, before augmentation.

textures, from each of the recorded videos approximately one frame every 1.5 seconds was grabbed. These images were then cropped to a resolution of 256×256 pixels and changed to greyscale with a pixel value between 0 and 1, in order to reduce computational cost and to avoid misidentification of phases due to colour instead of texture. The number of images generated for this study is shown in Fig. 3.

From Fig. 3 it can be seen that the dataset of the compound exhibiting the N_{TB} phase is not quite ideally balanced, with the respective phase representing approximately 1000 images more than the standard nematic phase. However, according to a study[43] where class imbalances were investigated in detail, it was pointed out that imbalances of the order of $2:1$ are not of significant concern, but that such imbalances only have marketable effects on prediction accuracies when imbalances like $20:1$ are included. The imbalances of our dataset should thus only have minimal impact on the accuracy, although of course a balanced set of class images would obviously be better. This could be achieved by leaving out images from the over-represented classes, but this would lead to fewer training images, which would have a larger effect on the accuracy than the class imbalance.

The collected images were separated into training, validation, and test data subsets at an approximate ratio of $70:15:15$. For this separation to provide accurate results, it is important that the subsets have no overlap with each other, to prevent data leakage, which would inflate the accuracy. Images of the same phase coming from the same video were therefore not divided between the subsets and further shuffled to ensure randomness within each batch. Overall, and before augmentation, this procedure provided roughly balanced datasets of about 1500 images, which should provide reasonable accuracies, especially since the crystalline and the isotropic phase are very distinct, either showing features of typical cracks and simply a black image, which are easy to identify by the machine learning models. During the investigations, images were further subjected to different augmentations, which will be discussed in more detail below.

Machine learning algorithms

As in previous investigations, sequential and inception models were also used for this study. Each model was implemented using Google Colab[44] with the TensorFlow[45] and Keras libraries[46] in Python. ADAM optimisation[47] was used in all models as it runs natively in Keras and is computationally efficient with little memory requirement. Categorical cross entropy was used as the loss function for all models to quantify the dissimilarity between the predicted probabilities and the true labels. ReLU activation[48] was used on each layer, with the output layer using SoftMax activation.[49] The stride of the convolutional layers was set to 2, and the padding was set as to ensure the output size was the same as the input. In cases where overfitting was observed, dropout regularisation was used, set at 0.5. To maximise the accuracy of the machine learning model output, underfitting as well as overfitting needs to be avoided. The two models employed are schematically shown in Fig. 4.

For the inception models, Google's prebuilt inceptionV3[50] model was used, as its architecture has been fine-tuned by experts for the sole purpose of image identification. Inception V3 has been trained on ImageNet,[51] a large database of approximately 14 Mio. classified images. This iteration of Google's inception model is freely available. The model utilises batch normalisation, factorised 7×7 convolutions, average and max pooling layers, and, like the CNN models, SoftMax activation on the output. The pre-loaded version of InceptionV3 in Keras comes with the weights and biases found when training on the ImageNet database; these are turned off when training with the datasets from this study to avoid any unintentional bias in the predictions. The number of inception blocks will also be greatly reduced as the fully intact and trained architecture has approximately 25 Mio. parameters. With only four classes and about 5000 images in each dataset, the complete Inceptionv3 architecture would result in memorising each image, delivering high accuracies but making any predictions meaningless. At the end of each chosen number of inception blocks, a global average pooling layer is used as well as a SoftMax activation layer and dropout layer(s) when required.

Each model was trained for 50 epochs on both the training and validation datasets. The accuracy (Fig. 5(a)) and loss (Fig. 5(b)) of each epoch was monitored and once the initial training was complete, the learning curve was used to evaluate
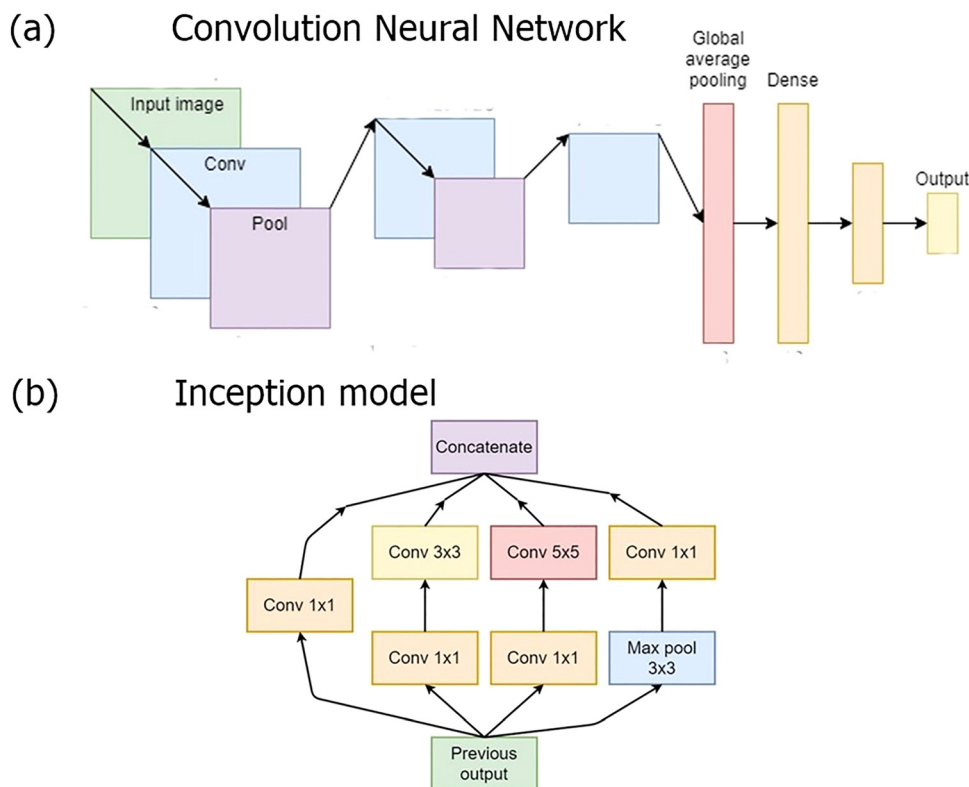
Fig. 4 General representation of (a) the convolutional neural network (CNN) model and (b) the Inception model, employed.
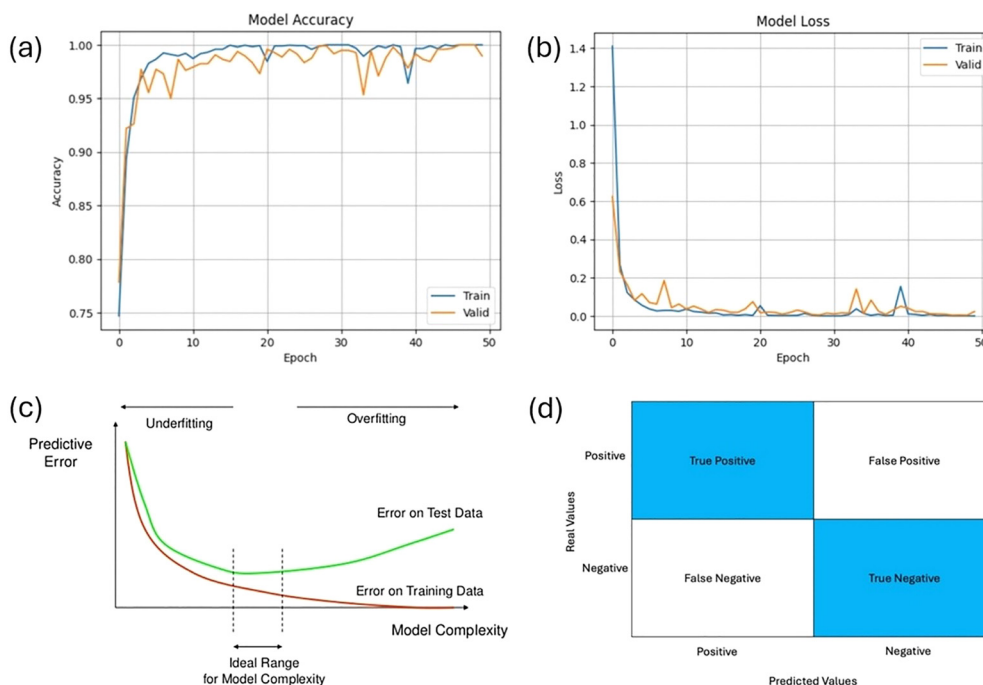

Fig. 5 Typical (a) accuracy and (b) loss curve of a 5-layer CNN with flip augmentation. (c) Schematic illustration of under- and overfitting to determine the ideal range of model complexity for which the machine learning model provides best predictions. (d) Typical definition of a confusion matrix.

a model's performance. Successful training was characterised by accuracy and loss curves that follow a similar pattern as the exemplary data depicted in Fig. 5(a) and (b), with the accuracy for the training and validation datasets converging at a similar value close to one and the loss curves converging at a low value close to zero. Overfitting is shown by diverging training and

validation curves, while underfitting is observed from low training loss at the beginning and dropping to an arbitrary minimum point (Fig. 5(c)).[52] The trained model would then be subjected to the test dataset of completely unseen images to evaluate model performance. Its predictions for each image were plotted on a confusion matrix to visualise the model's accuracy (Fig. 5(d)).

Hyperparameters and augmentations

The quality of learning curves and the model's efficiency are affected not only by the quality of the datasets (size and variation between training and validation) but also by the hyperparameters. Learning rate, batch size, dropout, and model size can be varied throughout testing to find the optimal implementation for each model with each dataset. A learning rate of 1×10^{-4} was chosen, which is sufficiently small to allow for the optimisation algorithm to find the minimum, but large enough to not significantly slow the training process.

The batch size has a direct impact on the accuracy of a model and its efficiency during the training process. The ideal batch size will vary depending on the size of the dataset in relation to the complexity of the model. As the datasets in this study are all of roughly equal size, the model's complexity was varied and the optimum batch size found for the first dataset will be used for all subsequent datasets. This was done by adding convolutional layers or inception blocks until the prediction accuracy starts to decrease or until satisfactory test accuracies between 90–100% were achieved. If similar results were observed in two models of differing complexity, the least complex model was chosen as the sufficiently optimal solution. A similar approach was applied to regularisation and dropout, adding a dropout layer to each successful iteration of a model and evaluating its effect on performance and accuracy.

In previous studies we have shown that flip augmentations are particularly effective in generating larger datasets without loss in phase prediction accuracy. Here, this was achieved both by manually editing images using batch editing software or by using the inbuilt augmentation layers from the Keras library during training. An investigation of the different types of augmentation, their effect on the dataset and the model's performance was conducted using the same approach as to testing the hyperparameters of the models. The most effective augmentation or combination of augmentations was then used on all datasets to improve the accuracy of the models.

Three augmentations were chosen for investigation: (i) brightness, (ii) contrast and (iii) flip augmentation. These were chosen in a way to significantly alter the appearance of the texture images without distorting the features key to identification. It should be mentioned that the Keras library also offers zoom and translation augmentation layers. These were not used as they were found to distort or change the image in undesirable ways. Using the zoom layer resulted in significant pixelation of images, which could result in the model being unable to identify certain key features. The translation layer applies random translations to each image during training, filling empty space with the part of the image that has been displaced. This generates boundaries which could result in some anomalous features and misinterpreted as being characteristic of a texture, thus leading to the inability to identify the actual phase.

Flip augmentations progress by flipping images on either one or both axes. In this study, we used a Keras augmentation layer for the CNN models which randomly selects images during each epoch and flips them depending on the conditions given by the user. Both horizontal and vertical flip were used to maximise the variation between the augmented images and the originals.

When inception models were employed, these are functional models rather than sequential, and as such, inception models cannot use the augmentation layer during training. All augmentations used for functional models have been completed manually using the BeFunky[53] batch editing software. Brightness and contrast augmentations were implemented using a Keras augmentation layer within a range of 0.2–0.8, avoiding either extreme of 1 or 0 where images lose all features, becoming either a blank white image or a blank black image. This protected against possible confusion during training, relating to false positives where darkened textures are misidentified as the isotropic phase.

Horizontal and vertical flipping was used for both, the CNN and the Inception models. For CNNs and inbuilt Keras function was used that selected a portion of images in each batch and applied the chosen flip augmentation. For the Inception models manual vertical flipping was applied to all the images in the training dataset, because the random function as used for the CNNs was not compatible with the Inception models within Keras. Further, due to the complexity of the Inception models it is beneficial for those datasets to have an increased size.

For improved computational performance all training and testing was completed in Google Colaboratory, a hosted Jupyter Notebook service that allows access to the GPUs hosted on Google servers. The Nvidia T4 Tensor Core GPU was used for all models as they have been specifically designed for machine learning and deep learning training.[54]

Results and discussion

Ferroelectric nematic compound NT3.5

Our machine learning investigation of liquid crystals with various nematic phases is divided into three main categories, (i) the phase sequence which includes a ferroelectric nematic phase, **NT3.5**, (ii) that which includes a twist-bend nematic phase, **CB6O.7**, and (iii) a hypothetical sequence that combines the two compounds, where we also distinguish between the two standard nematic phases, because these do exhibit quite different textures, as mentioned above. We thus also investigate in how far different textures of the same phase, here the standard nematic, can be distinguished. Further, we employ two different machine learning models, convolutional neural networks (CNN) and Inception models, varying a range of
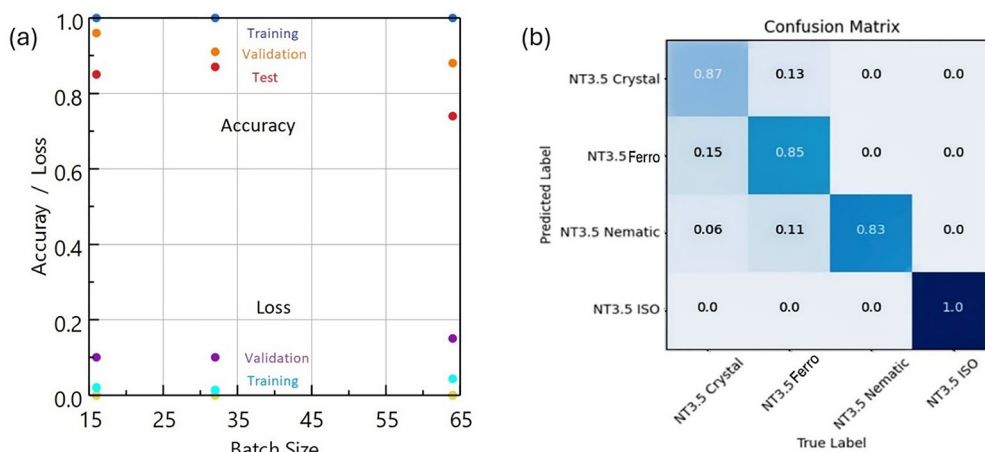
Fig. 6 (a) Accuracy and Loss for the training, validation and test cycles for a variation of the batch size for a single layer CNN. (b) Confusion matrix from testing with a single layer CNN and batch size 32. The dataset uses was that of the ferroelectric nematic compound **NT3.5**.

hyperparameters to unveil some general rules of thumb for texture characterisation *via* machine learning.

Using a basic CNN model with a single convolutional layer, one max pooling layer and a final dense layer to give output, testing was performed on some of the model's hyperparameters including batch size and learning rate. In the following, the effect of augmentations was investigated with different levels of complexity, followed by different regularisation techniques. Similarly, the inception model was studied.

The batch size, *e.g.* the number of data points given to a model at each iteration, can influence the learning of the model during training. The optimal batch size can depend on the size of the dataset, the optimisation algorithm used, or hardware constraints. As all the datasets in this study are of similar size and the same hardware was used throughout, an initial test was carried out to determine the optimal batch size, which was then used for the remainder of the investigations. It was found that a batch size of 32 gave the lowest validation loss and the highest test accuracy, together with the smallest amount of noise observed for the loss curves (Fig. 6). As such, for all further testing a batch size of 32 was to be used.

Augmentations were applied to the **NT3.5** dataset to artificially increase the number of images used for model training and to increase the variability between those images. This was implemented using a Keras augmentation layer that augments and randomly selects images in each batch, every epoch. Horizontal and vertical flip, brightness, and contrast augmentations were all individually tested and compared against a non-augmented dataset with increasing model complexity. It was found that increasing the model complexity resulted in a slight increase in the test accuracy of the flip-augmented dataset. Yet on the other hand, the brightness- and contrast-augmented datasets performed much below the models with the non-augmented dataset. The difference in test accuracies for each model is illustrated in Fig. 7. The graph clearly shows that brightness and contrast augmentations are unsuccessful augmentations, associated with large variations between individual runs during the test phase, as illustrated by large errors.

Even at higher levels of complexity, the models with brightness and contrast augmentations consistently displayed low validation and test accuracies with diverging losses. As the texture images of the datasets are greyscale, it is possible that
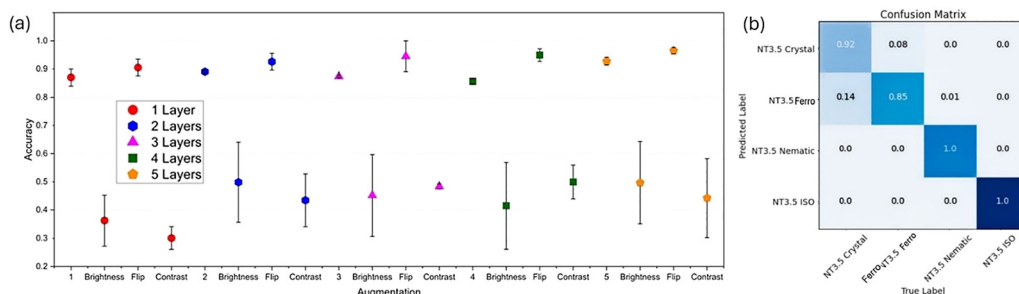

Fig. 7 (a) Effect of varying augmentations such as brightness, flip and contrast on the accuracy of the phase identification for increasing CNN complexity from 1 to 5 layers. Integers indicate the number of CNN layers used without augmentation. Only flip augmentation increased the model performance, while brightness and contrast resulted in poor performance. (b) Confusion matrix for the best performing CNN model only using flip augmentations. The dataset uses was that of the ferroelectric nematic compound **NT3.5**.
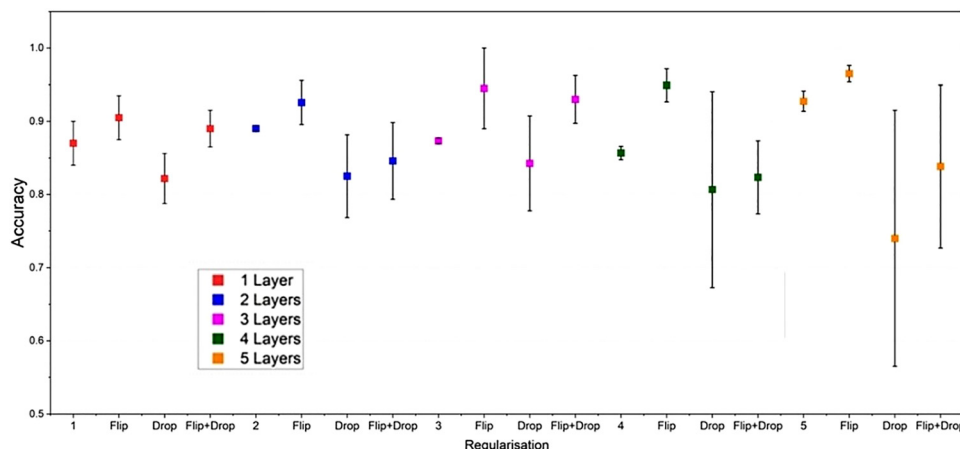
Fig. 8 Effect of regularisation on models with increasing complexity using 1–5 CNN layers. Integers indicate the number of CNN layers used without augmentation. Flip augmentation increases the accuracy as compared to non-augmented datasets, which both pure dropout and flip augmentation + dropout decrease the accuracy. It is thus evident that pure flip augmentation is the best process to increase model performance and accuracy.

even small changes to brightness and/or contrast largely obscure the features of the textures – the datasets with contrast tending to be identified as isotropic and the datasets with brightness tending to be categorised as crystalline. Horizontal and vertical flips were the only augmentation used in subsequent investigations.

A method to reduce possible overfitting is regularisation, done by reducing the weights put on connections between layers in the network or by removing them entirely; this latter case is known as dropout. To further increase the accuracy of the models used in this study, a dropout of 0.5 was used on models with both an un-augmented dataset and a flip-augmented dataset, essentially removing at random 50% of the connections between layers at random between each epoch.

Fig. 8 depicts the accuracies of each CNN model with flip augmentation, dropout, and both flip augmentation and dropout. The models with flip augmentation are the best performing and display an increase in accuracy with each added level of complexity. Adding a dropout layer to both the augmented and un-augmented datasets resulted in decreased accuracies, which varied significantly with each test, resulting in clearly larger errors. The best-performing model within this group of testing was the five-layer flip-augmented CNN with a test accuracy of 0.96 ± 0.01. One could anticipate that the even more complex models may lead to even higher accuracies, but this is generally not the case, due to overfitting. This can also be seen on the respective learning curves, where in general flip augmentation + dropout showed lower accuracies and higher loss, together with larger noise, when compared to pure flip augmentations. We therefore terminated this investigation at the 5-layer model.

Using the dataset of the ferroelectric nematic material NT3.5 we finally also employed a different machine learning model. For testing with the inceptionV3 model, the same NT3.5 dataset was used as before with manually implemented horizontal and flip augmentation. The complexity of the model is varied by altering the number of inception blocks. Due to the much larger number of variables, the inception model proves

successful with already the one-block model outperforming the five-layer CNN leading to a test accuracy of 0.99 ± 0.01. Increasing the complexity still further improves the test accuracy, which is most likely the result of overfitting, as suggested by the accuracy and loss curves. With this in mind, and with high accuracies being achieved by the two and three-block models, dropout layers were only added to the lowest performing one-block model, with the results depicted in Fig. 9 (note the change in scale as compared to previous CNN graphs).

Twist-bend nematic compound CB6O.7

The dataset of the compound **CB6O.7** with the twist-bend nematic phase is very comparable to the previous one in size. The same hyperparameters as used with the **NT3.5** dataset, were also used for the testing of **CB6O.7**, *e.g.* a batch size of 32 and no further augmentations with respect to brightness and contrast as these had a detrimental effect on the model's accuracy. A procedure very similar to that of the previous section was followed, starting with a CNN model with one convolutional layer, then introducing flip augmentation, and finally a layer of 0.5 dropout. The accuracies for increasing the model's complexity are depicted in Fig. 10.

As can be seen in Fig. 10, the test accuracies for the simple non-augmented models increase from about 89% to 96% as the model complexity is increased from one to three layers. Application of flip augmentations increases the average test accuracy slightly by another 1–2% to 97–98% prediction accuracy, while additional dropout not only decreases the overall accuracy considerably, but also increases the errors observed. Overall, the best performing model is the CNN with five layers and flip augmentation with an average of 98% accuracy.

For demonstration we also show the respective learning curves for the model accuracy and loss for the 1-layer, 3-layer and 5-layer CNN in Fig. 11(a)–(c), respectively. For the model accuracy it can clearly be seen that the validation curves approach the training curves as the CNN complexity increases. At about 40 epochs the training accuracy has reached
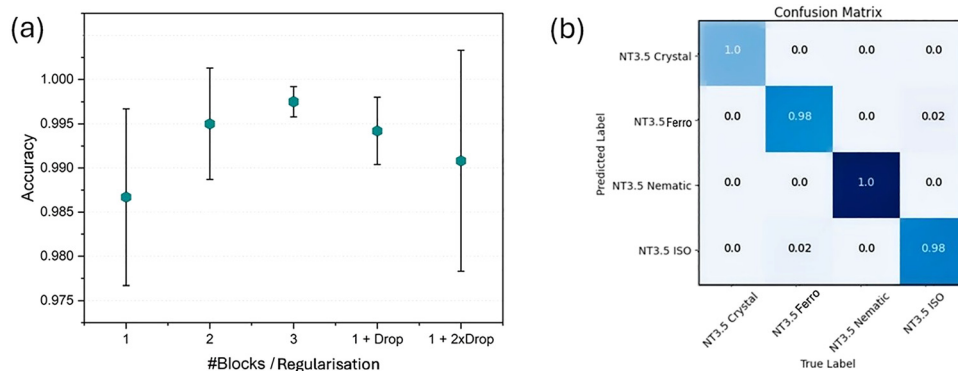
Fig. 9 (a) Accuracy as a function of inception model complexity with regularisation only applied to the simplest model with one inception block. (b) The confusion matrix verifies an excellent prediction of the phase sequence of the ferroelectric nematic compound **NT3.5**.
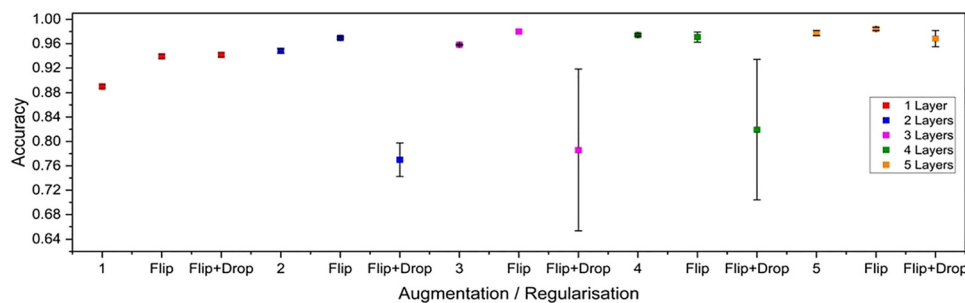

Fig. 10 Average test accuracies from CNN models with 1–5 layers, flip augmentation and flip + dropout for the **CB6O.7** dataset of the material exhibiting a twist-bend nematic phase in its phase sequence. Integers indicate the number of CNN layers used without augmentation.

approximately 98%, 99% and 100% for the 1-, 3-, and 5-layer model, while the validation curves reach 93%, 97% and 99%.

Similarly, the training loss curves are approximately 0.5%, 0% and 0% for the 1-, 3-, and 5-layer model, while the validation loss approaches these values to about 5%, 0.1%, and 0.05%. At the same time the noise on the loss curves is strongly reduced between the 1-layer and the 3-layer CNN. The behaviour is further evidenced in the confusion matrices of Fig. 12.

The confusion matrix obtained from the tests of this model shows that the increase in complexity enabled more accurate
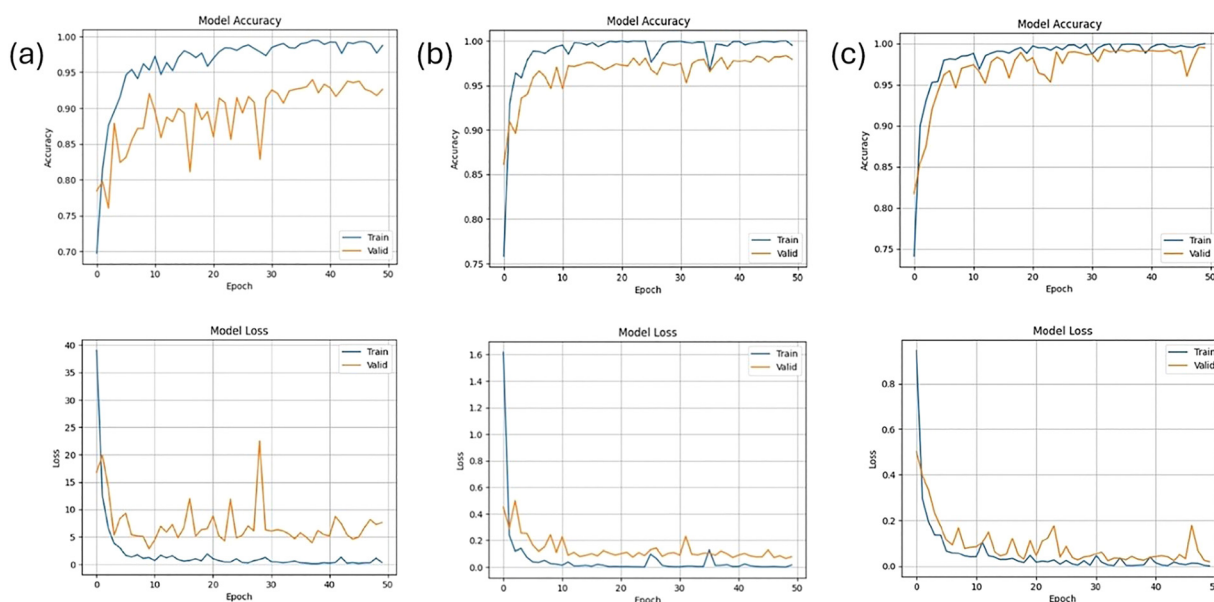

Fig. 11 Comparison of the learning curves for accuracy and loss for the (a) 1-layer, (b) 3-layer, and (c) 5-layer CNNs with flip augmentation.
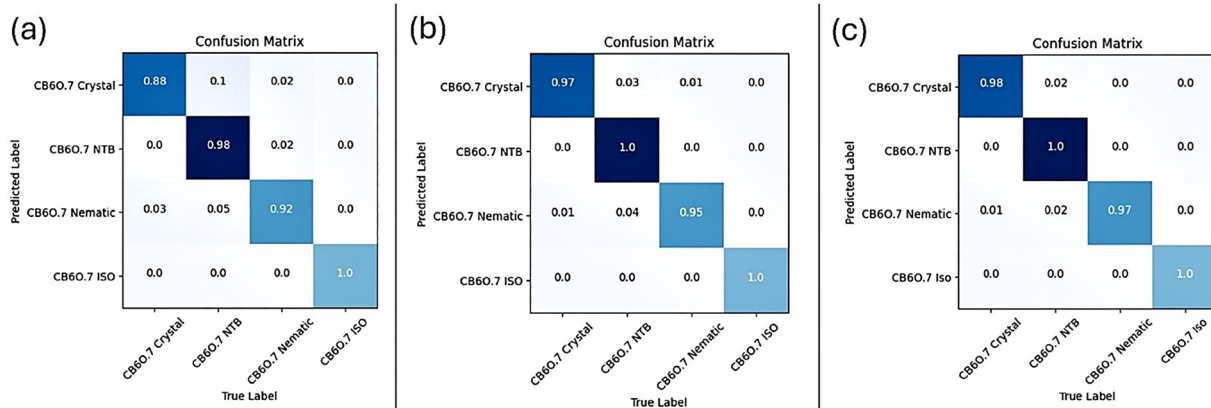
Fig. 12 Confusion matrices for the phase classification of **CB6O.7** with a twist-bend nematic phase, for a (a) 1-layer, (b) 3-layer, and (c) 5-layer CNN with flip augmentation.

identification of all phases of **CB6O.7**. The twist-bend phase showed an increase in accuracy to almost 100%. The greatest improvement is seen for the crystalline phase, for which the test accuracy increased from approximately 88% to 98%.

Using the same **CB6O.7** dataset, the InceptionV3 model proved immediately adept at identifying the phase sequence. With only one inception block, test accuracies of 0.9992 ± 0.0008 were achieved (Fig. 13). Manual flip augmentation was used in all cases.
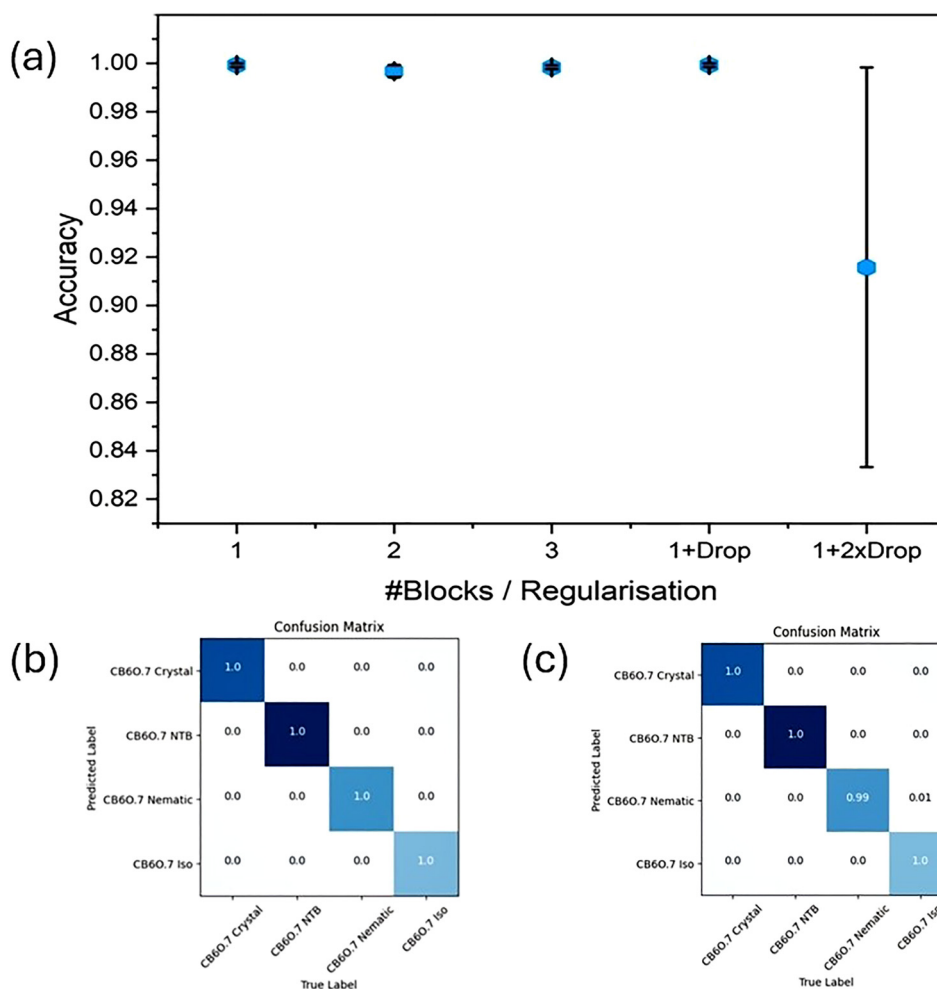

Fig. 13 (a) Average test accuracies of inception models tested with the **CB6O.7** dataset. (b) Confusion matrix from a one-block inception model and (c) confusion matrix for a 1-block inception model with one layer of dropout.

It appears that any number of inception blocks up to at least three would provide an accuracy of very close to 100%, yet a closer look at the learning curves shows that already after few epochs the training and validation curves have reached an accuracy of 100% at zero loss, even for the one-block model. In principle this implies that the inception model is simply too complex for the task. The addition of dropout regularisation, an action that should help to avoid overfitting, even decreases the accuracy slightly. Similar to the compound **NT3.5** with the ferroelectric nematic phase, more so for **CB6O.7** with the twist-bend nematic phase, the inception models are too complex for the task of phase sequence characterisation to exclude overfitting, which will always lead to accuracies of roughly 100%. It can thus be suggested that the best model employed for tasks as discuss so far are 3-layer CNNs with flip augmentation.

Combined dataset of nematic NT3.5 and CB6O.7

To increase the complexity of the characterisation task, we combined the textures of the ferroelectric nematic **NT3.5** and the twist-bend nematic **CB6O.7**, while also treating the two standard nematic phases as independent, due to the fact that they exhibited different texture appearances throughout, as mentioned above. This purely nematic dataset describes a hypothetical nematic phase sequence with dataset size of about 6800 images.

Fig. 14(a) demonstrates that an increase in the model's test accuracies is related to the number of convolutional layers. In all cases, the inclusion of flip augmentation resulted in higher accuracies at each level of complexity. The inclusion of a

0.5 dropout layer to the augmented datasets resulted in lower test accuracies (Fig. 14(b)). The exemplary confusion matrix shown in Fig. 14(c) shows that the CNN model is clearly able to identify both the ferroelectric nematic and nematic twist-bend phases but is slightly less effective at identifying the standard nematic phases. Nevertheless, with both accuracies for the standard nematic phase being clearly above 90%, it is demonstrated that machine learning models are feasible to identify not only different phases but also different textures of the same phase.

Unlike it was observed with the previous datasets, training the 1-block inception model with the "all nematic" dataset did not result in 100% accuracy. The highest test accuracy (0.998 ± 0.000) was achieved by a 3-block model with manual flip augmentation applied. Fig. 15(a) shows that the inclusion of a 0.5 dropout layer resulted in a decrease in test accuracies within the limits of error.

Fig. 15(c) shows the learning curves for the one-block inception model with flip augmentation. Training and validation accuracies start at high levels from the first epoch and reach a value close to 100% by the end of the 50 epochs. Applying this model to the test dataset, accuracies of 0.987 ± 0.003 are achieved. The associated confusion matrix (Fig. 15(b)) confirms this behaviour, with the lower accuracies being found when identifying the standard nematic phases. However, these two phases are not often mistaken for one another despite being the same phase.

When comparing the highest accuracies achieved by the CNN and Inception models for the all nematic dataset (0.970 ± 0.003 *vs.* 0.987 ± 0.003, respectively), the inception models
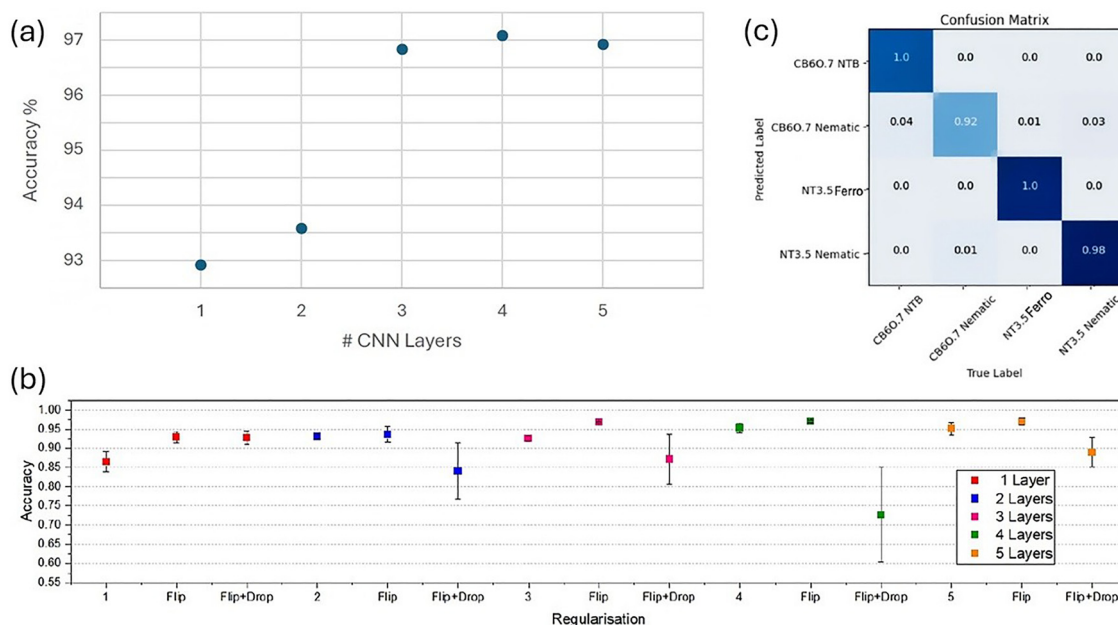

Fig. 14 (a) An increasing CNN model complexity leads to an increase in prediction accuracy until a plateau is reached for about three CNN layers. (b) Flip augmentation slightly increased the model accuracy, while dropout regularisation exhibited a rather detrimental effect on the accuracy. Integers indicate the number of CNN layers used without augmentation. (c) Confusion matrix for the different nematic phases/textures of the hypothetical nematic phase sequence for the 4-layer flip-augmented CNN model.
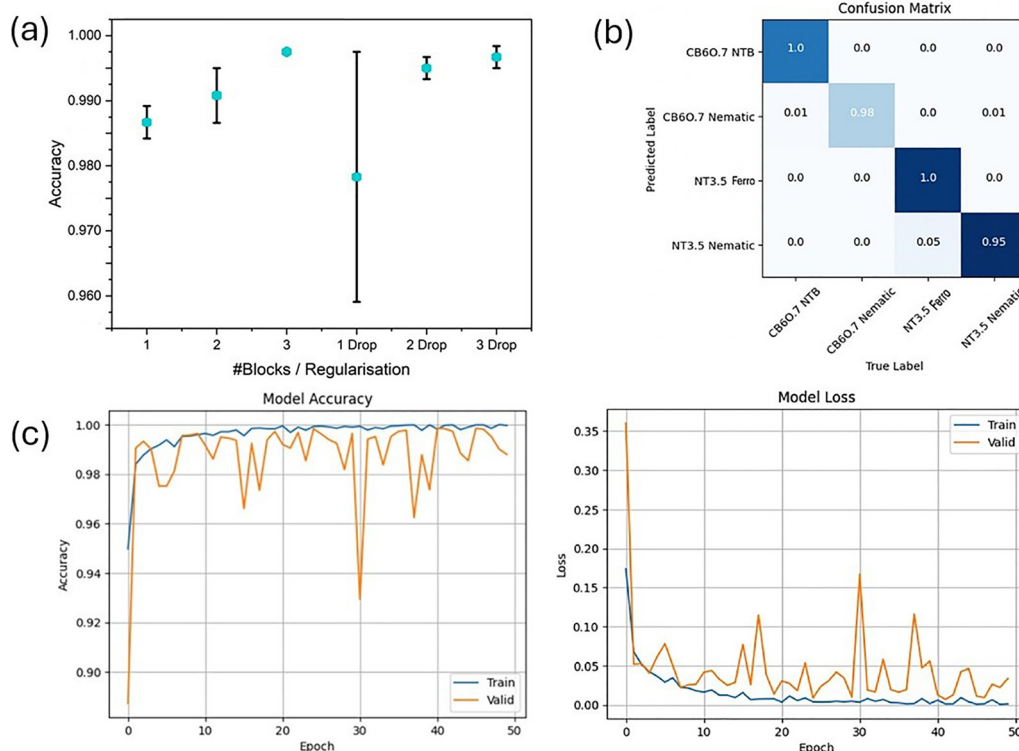
Fig. 15 (a) Average accuracies of the inception models with increasing complexity, *e.g.* number of inception blocks, without and with dropout regularisation. (b) Confusion matrix and (c) learning curves for the 1-block inception model with flip augmentation.

appear slightly more adept at identifying the LC phases from their texture images. It is clear, however, that the CNN models offer sufficient accuracies to demonstrate the feasibility of such models as well. Given the fact that training an inception model consumes considerably more time and computer resources than training a CNN, the latter is certainly sufficient for characterisation, at least for unconventional nematic phases. The effect of adding dropout to these models rarely had a positive impact on accuracy.

Conclusions

Sequential CNNs and modified pre-built Inception models were tested for their ability to identify unconventional nematic liquid crystal phases. Three datasets, including ferroelectric nematic and twist-bend nematic phases besides the standard nematic, were tested with different machine learning architectures for their suitability. In this process the model complexity was varied to assess its effect on accuracy as well as the use of various augmentation and regularisation techniques. The use of brightness and contrast augmentation led to the loss of textural information, resulting in considerably lower test accuracies and higher errors. Flip augmentation was the most effective means and resulted in higher test accuracies when compared to non-augmented datasets.

Without loss of generality, flip-augmented three- to four-layer CNNs were found to be of sufficient complexity to

fulfil the task of characterising all sequences to better than 95%. The inception model was found to achieve higher accuracies of 99% with as little as one inception block, however, learning curves during training and validation suggested that these high accuracies were most likely the result of overfitting. It is worth noting that for both models the inclusion of dropout regularisation resulted in the worst test accuracies. Thus, the inception model is deemed to be far too complex for the datasets investigated, even with considerable regularisation employed. Inception models are also computationally much more expensive than CNNs. One can thus conclude that for the present investigation the use of Inception models is not necessary or justified.

The datasets used in this study are relatively small in machine learning terms and exhibit minor class imbalances. While these problems do not necessarily negatively impact the efficacy of the findings, greater accuracies could possibly be achieved in cases with larger and more balanced datasets. If such higher accuracies are required rather than the ones achieved, it is clearly necessary to collect larger and more balanced datasets. In the latter case, one may then also need to resort to more complex machine learning models at the expense of computational costs.

Conflicts of interest

There are no conflicts to declare.

Data availability

All the important data is shown in the paper. Raw data, such as images used for training, can be made available upon reasonable request.

References

1 D. Demus and L. Richter, *Textures of Liquid Crystals*, Verlag Chemie, Weinheim, 1978.

2 G. W. Gray and J. W. Goodby, *Smectic Liquid Crystals – Textures and Structures*, Leonard Hill, Glasgow, 1984.

3 I. Dierking, *Textures of Liquid Crystals*, Wiley-VCH, Weinheim, 2003.

4 I. Dierking, *Liq. Cryst.*, 2025, DOI: 10.1080/02678292.2025.2517865.

5 E. M. Barrall, R. S. Porter and J. F. Johnson, *J. Phys. Chem.*, 1967, **71**, 895.

6 L. V. Azoroff, *Mol. Cryst. Liq. Cryst.*, 1980, **60**, 73.

7 A. Piven, D. Darmoroz, E. Skorb and T. Orlova, *Soft Matter*, 2024, **20**, 1380.

8 H. Y. D. Sigaki, R. F. de Souza, R. T. de Souza, R. S. Zola and H. V. Ribeiro, *Phys. Rev. E*, 2019, **99**, 013311.

9 H. Y. D. Sigaki, E. K. Lenzi, R. S. Zola, M. Perc and H. V. Ribeiro, *Sci. Rep.*, 2020, **10**, 7664.

10 A. A. B. Pessa, R. S. Zola, M. Perc and H. V. Ribeiro, *Chaos, Solitons Fractals*, 2022, **154**, 111607.

11 C.-H. Chen, K. Tanaka and K. Funatsu, *Mol. Inf.*, 2019, **38**, 1800095.

12 Huaizhe Yu Yankai Cao and V. M. Zavala N.L. Abbott, *ACS Sens.*, 2018, **3**, 2237.

13 M. Walters, Q. Wei and J. Z. Y. Chen, *Phys. Rev. E*, 2019, **99**, 062701.

14 E. N. Minor, S. D. Howard, A. A. S. Green, M. A. Glaser, C. S. Park and N. A. Clark, *Soft Matter*, 2020, **16**, 1751.

15 I. Dierking, J. Dominguez, J. Harbon and J. Heaton, *Liq. Cryst.*, 2023, **50**, 1526.

16 I. Dierking, J. Dominguez, J. Harbon and J. Heaton, *Front. Soft. Matter*, 2023, **3**, 1114551.

17 N. Osiecka-Drewniak, Z. Galewski and E. Juszyńska-Gałązka, *Crystals*, 2023, **13**, 1187.

18 N. Osiecka-Drewniak, A. Deptuch, M. Urbanska and E. Juszynska-Gałązka, *Soft Matter*, 2024, **20**, 2400.

19 R. Betts and I. Dierking, *Soft Matter*, 2024, **20**, 4226.

20 R. Betts and I. Dierking, *Soft Matter*, 2023, **19**, 7502.

21 I. Dierking, J. Dominguez, J. Harbon and J. Heaton, *Liq. Cryst.*, 2023, **50**, 1461.

22 N. Osiecka-Drewniak, M. Piwowarczyk and M. Gałązka, *J. Mol. Liq.*, 2025, **428**, 127511.

23 L. A. Madsen, T. J. Dingemans, M. Nakata and E. T. Samulski, *Phys. Rev. Lett.*, 2004, **92**, 145505.

24 B. R. Acharya, A. Primak and S. Kumar, *Phys. Rev. Lett.*, 2004, **92**, 145506.

25 H. F. Gleeson, S. Kaur, V. Goertz, A. Belaissaoui, S. Cowling and J. W. Goodby, *ChemPhysChem*, 2014, **15**, 1251.

26 M. Cestari, S. Diez-Berart, D. A. Dunmur, A. Ferrarini, M. R. de la Fuente, D. J. B. Jackson, D. O. Lopez, G. R. Luckhurst, M. A. Perez-Jubindo, R. M. Richardson, J. Salud, B. A. Timimi and H. Zimmermann, *Phys. Rev. E:Stat., Nonlinear, Soft Matter Phys.*, 2011, **84**, 031704.

27 D. Dunmur, *Crystals*, 2022, **12**, 309.

28 R. J. Mandle, *Molecules*, 2022, **27**, 2689.

29 R. J. Mandle, *Soft Matter*, 2016, **12**, 7883.

30 M. Szmigielski, *Soft Matter*, 2023, **19**, 2675.

31 C. Meyera, I. Dozov, P. Davidson, G. R. Luckhurst, I. Dokli, A. Knezevic and A. Lesac, *Proc. SPIE*, 2018, **10555**, 105550Z-1.

32 X. Chen, E. Korblova, D. Dong, X. Wie, R. Shao, L. Radzihovsky, M. A. Glaser, J. E. Maclennan, D. Bedrov, D. M. Walba and N. A. Clark, *Proc. Natl. Acad. Sci. U. S. A.*, 2020, **117**, 14021.

33 O. D. Lavrentovich, *Proc. Natl. Acad. Sci. U. S. A.*, 2020, **117**, 14629.

34 N. Sebastián, M. Copic and A. Mertelj, *Phys. Rev. E*, 2022, **106**, 021001.

35 E. Cruickshank, *ChemPlusChem*, 2024, **89**, e202300726.

36 M. A. Osipov, *Liq. Cryst. Rev.*, 2024, **12**, 14.

37 R. Kumar Khan, *Appl. Res.*, 2025, **4**, e70018.

38 R. Walker, *Liq. Cryst. Today*, 2020, **29**, 2.

39 R. J. Mandle, *Liq. Cryst.*, 2022, **49**, 2019.

40 R. Walker, D. Pociecha, G. J. Strachan, J. M. D. Storey, E. Gorecka and C. T. Imrie, *Soft Matter*, 2019, **15**, 3188.

41 N. Tufaha, E. Cruickshank, D. Pociecha, E. Gorecka, J. M. D. Storey and C. T. Imrie, *Chem. Eur. J.*, 2023, **29**, e202300073.

42 VLC media player, (2006), VideoLan, [Online], Available: https://www.videolan.org/.

43 M. Buda, A. Maki and M. A. Mazurowski, *Neural Networks*, 2018, **106**, 249.

44 Available at: https://colab.google/.

45 Available at: https://www.tensorflow.org/.

46 Available at: https://keras.io/.

47 D. P. Kingma and J. B. Adam, *arXiv*, 2014, preprint, arXiv:1412.6980, DOI: 10.48550/arXiv.1412.6980.

48 A. F. Agarap, Deep learning using rectified linear units (relu), *arXiv*, 2018, preprint, arXiv:1803.08375, DOI: 10.48550/arXiv.1803.08375.

49 Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition, in *Neurocomputing: Algorithms, Architectures and Applications, NATO ASI Series (Series F: Computer and Systems Sciences), 1989*, ed. J. S. Bridle, F. F. Soulié and J. Hérault, Springer, Berlin, Heidelberg, 1990, vol. 68, pp. 227–236, DOI: 10.1007/978-3-642-76153-9_28.

50 C. Szegedy, V. Vanhoucke, S. IoNe, J. Shlens and Z. Wojna, Rethinking the Inception Architecture for Computer Vision, *arXiv*, 2015, preprint, arXiv:1512.00567, DOI: 10.48550/arXiv.1512.00567.

51 J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, 2009, pp. 248–255, DOI: 10.1109/CVPR.2009.5206848.

52 H. N. K. Al-Behadili, K. R. Ku-Mahamud and R. Sagban, Rule pruning techniques in the ant-miner classification algorithm and its variants: A review, in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, Penang: IEEE, 2018, pp. 78–84, DOI: 10.1109/ISCAIE.2018.8405448.

53 T. Tatar, BeFunky – Batch Photo Editor, [Online], Available: https://www.befunky.com/create/batch-photo-editor/.

54 Z. Jia, M. Maggioni, J. Smith and D. P. Scarpazza, Dissecting the NVidia Turing T4 GPU via Micro-benchmarking, *arXiv*, 2019, preprint, arXiv:1903.07486, DOI: 10.48550/arXiv.1903.07486.