Bespoke data augmentation and network construction enable developmental morphological classification on limited microscopy datasets

Ian Groves[1,2], Jacob Holmshaw[1], David Furley[1,2], Matthew Towers[2], Benjamin D. Evans[3], Marysia Placzek[2], Alexander G. Fletcher[1,*]

[1] School of Mathematics and Statistics, University of Sheffield, Hicks Building, Hounsfield Road, Sheffield, S3 7RH, UK

[2] School of Biosciences, University of Sheffield, Firth Court, Western Bank, Sheffield, S10 2TN, UK

[3] Department of Informatics, School of Engineering and Informatics, University of Sussex, Falmer, Brighton, BN1 9RH, UK

[*] E-mail: a.g.fletcher@sheffield.ac.uk

ABSTRACT

Deep learning is a powerful tool for image classification, yet training deep learning models often requires large datasets, which are not always available in biological research. Here we use a small dataset to train an accurate classifier by employing data augmentation regimes and Bayesian optimisation. We establish a new staging system for the Hamburger-Hamilton stage 10 chick embryo brain, based on morphology, identifying three sub-stages uncoupled from somite number, the usual metric used to stage chick embryos. Incorporating biologically informed, data-driven preprocessing, we train a deep convolutional neural network to classify our sub-stages. We find that augmenting our images with a combination of rotation, blur, shear, and cutout transformations is effective in training an accurate classifier on a small microscopy dataset. We then apply our approach to an unrelated small microscopy dataset comprising images of limb buds, finding good agreement with the results from brain classification. Through saliency analysis, we ensure that relevant features are being used by our classifier, gain insight into the efficacy of different data augmentations, and identify sub-stage specific features. In summary, we apply biological domain expertise to design effective data augmentation regimes for training a neural network-based image classifier on small, specialised datasets, classifying the developing brains and limbs with up to 90.9% and 94.4% accuracy respectively. These reliable classifiers can inform future experiments.

INTRODUCTION

Deep learning (DL) is increasingly becoming embedded in biomedical research (1). Deep neural networks (DNNs) have been used to powerful effect in, for example, classifying histological samples (2), and nucleus segmentation (3). A common maxim for training DNN-based image classifiers is that performance improves with increasing amounts of training data (4,5). However, in many areas of biomedical science, it is often impractical to generate large amounts of data due to cost, time, or ethics. In such cases, it is important to maximise the utility of limited data when training DNNs, but how best to do this remains an open question. Here we address this in the context of microscopy images used for developmental biology research.

The use of DNNs in studies of *in vitro* fertilisation and the cell cycle (6) first highlighted the power of this approach in developmental biology. In this field, common challenges are to determine how far an embryo has advanced along its developmental trajectory: the embryo's 'developmental stage', or to compare embryos that have been subjected to different experimental regimes. DNNs were recently used to classify the developmental stage of zebrafish tailbuds (an experimental model for posterior spinal cord growth) (7). In this study, networks were trained on 2 and 3-D image datasets, Bayesian optimization was then used to tune network architecture, and success was judged through classification accuracy. While this study provides proof-of-principle for using DNNs to classify developmental stages, it did not test the DNN's ability to classify unseen or out-of-distribution data – its capability to generalise (8). This aspect of DNNs in developmental biology is therefore yet to be fully leveraged, and an important goal is to determine the approaches that best improve generalisation.

Most developmental datasets are far smaller than those used routinely to train image classifiers e.g. ImageNet (9). When large datasets are not available, augmentation (through transformations such as rotations) is a particularly useful technique for expanding the training data, such that a classifier learns the class-defining features of the image, and disregards irrelevant features, including image acquisition artefacts (10). Augmentations are typically 'label-preserving', i.e preserve the class-defining features of the image. Additionally, new partially label-preserving augmentation methods have been developed. These include 'cutout', where a random portion of the image is masked (11), reducing the classifier's

reliance on those masked features and Möbius transformations, which are bijective conformal mappings that preserve angles (12) and may be effective in accounting for user error in microscopy image acquisition e.g. sample damage during preparation. Such partially label-preserving transformations may normalise the user error feature space and discourage the network from learning erroneous features, making the network more robust and less sensitive to artefacts in the dataset.

Besides data augmentation, classifier performance can be improved by optimising parameters that are set prior to training. These so-called hyperparameters include the number of computational units in the network, and the rate at which these units update – the learning rate. Hyperparameters are typically optimised via systematic (13) or random (14) search. Bayesian optimisation techniques are increasingly used, with a probability model informing which values to test (15). An open question then, when training DNNs on microscopy images, is how to exploit data augmentation and network optimisation techniques to suit small datasets?

Finally, saliency maps show which features of an image are used by the network to make a classification (16), and so reveal both the efficacy of data augmentation, and a DNN's capability to generalise. In this way, saliency maps provide insight into the 'black box' approach of using a high accuracy DNN classifier, an important step in increasing uptake of DNN use in developmental biology.

Here we asked whether these approaches can be harnessed to train a DNN that accurately classifies limited image datasets of developing chicks, and can be generally applied to other developmental microscopy datasets. Chick embryos have been used to drive our understanding of many aspects of early development (e.g. 17), but are difficult to obtain in large numbers. In the chick, the rate of development is sensitive to both intrinsic and extrinsic factors (for instance, temperature) so developmental staging is based on morphology, not chronological age. Classically, developmental stages are designated according to the Hamburger-Hamilton (HH) 1955 staging guide, which assigns embryos to stages HH1 (newly incubated egg) through 46 (newly hatched chick) (18). This staging guide is based on morphological metrics, but for early stages.relies on the number of somites – segmented tissue that forms at regular intervals on each side of the developing spinal cord (19). However, careful observational studies show that the rate/timing of development of

different body structures, including the brain – is subtly uncoupled from somite number, and increasingly, researchers are developing staging classifiers for distinct embryonic regions (19–23). Furthermore, molecular studies show that enormous changes can occur in a tissue over a single HH developmental stage. For instance, recent studies show significant changes in ventral parts of the brain - where the eyes and hypothalamus are developing - over HH10, a period that encompasses ~5 hours (28,29). While expert developmental neurobiologists are able to sub-stage the HH10 brain based on its broader morphology, those with less experience are unable to do so with confidence. Such accurate subclassification is imperative for future experiments.

Here, we sought to leverage our domain expertise to develop a reproducible, rapid, and automatable strategy for training deep convolutional neural network (DCNN) based image classifiers that performs well on small microscopy datasets. First, we classified subdivisions of the HH10 chick brain. We then tested both the efficacy of widely used and bespoke network architectures, and a large variety of data augmentations, to train a classifier that accurately classifies a limited, noisy microscopy dataset of the developing brain. Next, we confirmed the generality of the approach by applying the same strategy to a developing limb dataset. Our accurate classifiers (>90% accuracies) indicate that we have developed an important tool for analysing this type of dataset. Finally, our studies highlight how saliency analysis provides novel and unbiased insights into predictive biological features.

RESULTS

Somite number does not explain variation in HH10 brain morphology

Our visual inspection of images suggested that the HH10 chick brain could be divided into three sub-stages, according to the developing prosencephalon (forebrain) morphology, which we termed 10.1, 10.2 and 10.3 (**S1 Fig A**). At 10.1, the brain is kite-shaped, the optic vesicles (developing eyes) are shallow, and the angle of the prosencephalic neck, the anatomical landmark where the forebrain meets the posterior brain, is obtuse (**S1 Fig A, B iii**, left columns). At 10.2, the brain is hammer-shaped, the optic vesicles are more pronounced, and the posterior prosencephalon-prosencephalic neck are aligned (**S1 Fig A**, **B iii**, middle columns). At 10.3, the angle of the prosencephalon-prosencephalic neck is

acute, with a significant curving of the optic vesicles (**S1 Fig A, B iii**, right columns). The overall shape profiles used for our sub-staging are schematised in **S1 Fig B.**

We next investigated the relationship between our sub-stages and the classic Hamburger-Hamilton staging chart (18), where developmental stage is determined by somite number: HH10 embryos have 9, 10 or 11 somites, and are traditionally termed HH10- (9 somites), HH10 (10 somites), or HH10+ (11 somites) (18). First, we quantified somite number at each sub-stage (**Fig 1A**). We observed that each sub-stage (10.1, 10.2, 10.3) can have a wide range of somites (8-12 somites) (**Fig 1B**). We also found significant differences in somite number between 10.1 and 10.3 (*** p=0.0002), and between 10.2 and 10.3 (* p=0.015), but found no significant difference in somite number between 10.1 and 10.2 (p=0.079) **(Fig 1B)**. In conclusion, somite number is a poor predictor of chick HH10 brain development, underscoring the need for a finer-grained classification approach.

Unsupervised learning via *k*-means clustering supports three sub-stages, but performs poorly as a classifier

We next examined whether the choice of three sub-stages, motivated by our expert visual inspection, could be supported through clustering approaches. Inspired by an approach used to classify cell types in histology images (24), we combined principal component analysis (PCA) and *k*-means clustering. Our aim was to derive an appropriate number of classes for further supervised classifiers and test the efficacy of unsupervised classifiers.

A dataset of 151 images of chick brains was labelled as per reference images (**S1 Fig A,** 54 at 10.1, 55 at 10.2, and 42 at 10.3). We then used PCA for dimensionality reduction, to ensure comprehensible and robust clustering (25). To empirically determine an appropriate number of PCs, we generated a scree plot (**Fig 2A**), calculating the total variance explained by reducing our chick embryo sub-stage dataset into increasing numbers of PCs. We found that two PCs explained 81% of the dataset variance, and increasing the number of PCs further led to substantially diminishing returns.

We next clustered our data via *k*-means. To determine the number of clusters to use, we used the elbow method (**Fig 2B**): we clustered the dataset, with between 1 to 10 clusters, and selected the ideal number of clusters as the inflection point in the within-cluster sum of

squares (WCSS), where increasing *k* showed diminishing returns in WCSS. This point was reached at *k* = 3, which we then visualised (**Fig 2C**). We found substantial differences in the number of images within each cluster. Between the three clusters, clusters 1-3 contained 84, 18, and 49 images respectively (**Fig 2D**). We assumed that the sub-stage most represented in each cluster is the true classification for that cluster. Thus, sub-stages 10.1, 10.2 and 10.3 corresponded to clusters 2, 1 and 3, respectively (**Fig 2D**). Therefore, the classification accuracies achieved by PCA followed by *k*-means clustering were: 10.1: 38.9%; 10.2: 41.7%; 10.3: 44.9%.

As *k*-means clustering is unsupervised, it is unlikely to group data similar to a human observer. Looking beyond the classification accuracies, the distribution of images between clusters is skewed. Our '10.3' cluster, cluster 3 performed best, but the accuracy and the image distribution in the clusters suggest poor efficacy for classifying sub-stage (**Fig 2D**). This is likely because feature(s) are being used for clustering which are irrelevant to the true class. Whilst we confirmed that three sub-stages is an appropriate number of sub-stages to use, we next aimed to substantially improve the classification accuracy achieved by using DCNNs.

A DCNN trained using the ResNet50 architecture achieves a classification accuracy of up to 75% using a combined data augmentation regime of rotation & Gaussian blur

Initially, to reduce overfitting in our relatively small dataset (comprising 151 images), we tested supervised simple classifiers: random forest, support vector machine (SVM), and *k*-nearest neighbour (KNN). However, we were not able to train a sufficiently accurate classifier, with the highest individual and highest average validation accuracies achieved being 54.8% (RFC) and 38.3% (KNN) respectively (**S1 Table**).

We therefore opted to construct a strategy which would enable us to train a DCNN based classifier. Due to our limited datasets, we carefully considered each step of training and evaluating our DCNNs. First, we considered pre-processing. As our datasets consisted of both brightfield and phase-contrast microscopy images (bright and dark backgrounds respectively), we reasoned that a useful pre-processing step would be to normalise the histograms of each image.

Next, we augmented the dataset through transformations, which both expanded the number of datapoints for training, and normalised skewed image features unimportant for classification. For instance in microscopy images, subject orientation often varies greatly. Henceforth, we refer to our dataset augmented with rotations as our baseline.

We then determined our cross validation approach. The widely used hold out validation, (where some amount of the dataset, often 20%, is unavailable for training), was deemed unsuitable for our purposes, as with a limited dataset it is key to use every data point possible for training. We verified this empirically, showing a large amount of variability between highest validation accuracy achieved by models trained on our brain dataset with hold out validation (**S4 Table**). Therefore, we used *k*-fold cross validation with $k = 10$, (i.e. 15-16 images per fold, using the whole dataset for training) for all subsequent training.

For training a DCNN classifier, we first tested training using the widely used network architectures from InceptionV3 (26) and ResNet50 (27). These two neural networks have achieved high classification accuracies on ImageNet, a large database of over 14M images. We aimed to understand to what extent these established architectures could be used with limited microscopy data. We then trained, using the InceptionV3 and ResNet50 architectures (**S2-3 Tables**), on a variety of augmentation regimes (**S2 Fig**). Generally, InceptionV3 performed poorly, with similar averages across various data augmentations, and a maximum validation accuracy of 60.5%. This is better than *k*-means, which was 44.9% (**Fig 2D**), however the average validation accuracies achieved by InceptionV3 are in the range of 44-46%, so that overall, performance is similar to that achieved through *k*-means. With ResNet50 training, we saw a marked improvement in validation accuracies over InceptionV3 and the *k*-means clustering accuracy across all augmentations (except for crop, which only improved by an average of 3%). Our baseline & Gaussian blur augmentation regime (hereafter referred to as Gaussian blur) had the highest average validation accuracy, (69.2%), and highest fold accuracy (75.9%). Additionally, this regime achieved the second lowest standard deviation, an important metric in light of a limited dataset (**S3 Table**). Taken together, these results suggest that training with commonly-used image classification architectures is not effective in classifying small microscopy datasets. However this underperformance was substantially ameliorated by data preprocessing, and the Gaussian blur regime.

A bespoke neural network classifies developmental sub-stages of a small dataset of chick embryo brains with up to 90.9% accuracy

ResNet50 has been used successfully for image classification on datasets other than ImageNet, for example on ChestX-ray14, a dataset of X-ray images (28). ChestX-ray14 is substantially larger than our dataset, comprising 112,120 images. Therefore, considering that we did little fine-tuning of the hyperparameters of ResNet50, and that ResNet50 was constructed for large datasets, ResNet50 performed surprisingly well (up to 75.9%) when trained on our data. However, this performance leaves room for improvement so we set out to understand whether a bespoke network would achieve even higher classification accuracies. We aimed to incorporate domain expertise, by specifically considering the types of problems encountered in developmental biology datasets.

We constructed a model with a wide, VGG-16 block-style architecture (**S3 Fig**), which has been successful in image classification (29). We used Bayesian optimisation (**S6 Table**) and examples from the applied deep learning literature to tune the hyperparameters. We then determined the most useful and robust augmentation regimes (**Table 1**). Our baseline usually performed well, surpassing our best ResNet50 results with an average validation accuracy of 76.2%. However, for each fold, one of the various augmentations always had higher validation accuracy than the baseline. The poorest performing augmentation was baseline & crop, with an average validation accuracy of 47.1%. The best performing individual augmentation was baseline & Gaussian blur, with the highest average validation accuracy of 80.5% and the lowest standard deviation (excluding crop, which performed substantially below baseline). Baseline & cutout provided the best individual fold validation accuracy, of 91.7% at fold 9 (vs. Gaussian blur 88.9%). Despite baseline & cutout achieving the highest individual fold validation accuracy, we conclude that baseline & Gaussian blur overall performed best, as it achieved the highest validation accuracy on folds 4-7.

We also tested the efficacy of Möbius transformations, which have proven successful in other contexts (12) but are untested for microscopy image classification. We reasoned that Möbius transformations could introduce the DCNN to common microscopy artefacts, e.g. tissue bending during sample preparation. However, we found that our baseline and Möbius transforms performed more poorly than the baseline alone, and that sparse inclusion of Möbius transformations also decreased validation accuracy (**S5 Table**).

Our cross validation (**Table 1**) shows dataset partitions that lead to low accuracies, indicating that these partitions contain images with features that are not shared with the images in higher scoring partitions. These are likely artefacts introduced in chick dissection/preparation or imaging, e.g. due to tearing of the fragile tissue. As such, we sought to mitigate this invariance. Whilst the good performance of our baseline is intuitive, both the shear and Gaussian blur regimes also achieved high validation accuracies. For shear, we reasoned that this could reflect increasing the robustness of the network by introducing it to distortions similar to the types of tissue deformations routinely introduced in chick dissection/preparation or imaging Similarly, Gaussian blur could increase robustness by introducing the DCNN to data that resembles out-of-focus images, a common microscopy artefact.

To test whether Gaussian blur is having this effect, we applied the inverse operation by sharpening our images during preprocessing before our baseline augmentation (**Table 1**, 1 Sharpened). We found that sharpening increased average validation accuracy from 76.2% (baseline) to 78.2% (baseline sharpened). Further, we saw that poorly performing baseline folds (e.g. 2 and 8) had substantially higher validation accuracies. However, the results from sharpening as a preprocessing step do not entirely explain the discrepancy in validation accuracies between our baseline & Gaussian blur. This suggests that Gaussian blur is not only introducing the DCNN to out-of-focus images, but could be masking tissue tearing artefacts through smoothing of geometric lines or curves in the tissue structure. Moreover, these augmentations may be assisting the network to focus on a true sub-stage characteristic, rather than features arising from biological inter-sample variation. This suggests that these poorly scoring dataset partitions contain features that lead to an inaccurate classification, but this can be mitigated with reasoned preprocessing and augmentation.

Since different augmentations had scored highest on different folds, we next tested different combinations of these augmentations as an approach to reduce variability (**Table 1**, 1+4,5 RC, 1 + 2,4,5 RC). We identified an informed combined regime as a useful approach when dealing with high variability, small datasets. For example, folds 2 and 8 had the lowest average validation accuracies across all augmentations. For both, mixing the baseline with shear, Gaussian blur, and cutout resulted in substantially higher validation accuracies, allowing the network to learn ‘difficult’ features of our images.

In summary, we found training a bespoke neural network suits our problem, and that reasoned augmentations enhance network performance, achieving >90% validation accuracies, despite a small dataset.

Our bespoke neural network classifies a small dataset of developing chick limbs with up to 94.4% accuracy

We next set out to see if our pre-processing, architecture, and data augmentation approach could be applied to other small datasets. For this, we used previously published data comprising 269 images of developing chick limbs (30), separated into two labelled categories: 'control' (corresponding to the normal limb development), and 'treated', where growth had been transiently inhibited by the drug trichostatin A. We sought to determine whether our approach could be used to train a DCNN to classify control versus treated limbs. We set up the problem in a similar fashion to that used for brain classification, with minor changes as warranted by the switch from a multi-class to a binary classification problem. We made one addition to the baseline augmentation regime, which was to introduce flipped images along the horizontal axis, as the experimental protocol in (30) was to leave the left limb as control, and treat the right limb. We found that the validation accuracies achieved were generally higher than for the brain classification, especially across our baseline, which for folds 2, 5, 8 achieved the highest validation accuracies -- something not seen in the brain classification (**Table 2**). However, we found that the mixed regime performed best, with the joint highest average classification accuracy and lowest standard deviation of all the regimes tested. Additionally, the mixed regime only marginally underperformed the fold average once at fold 5. We therefore found that our strategy of reasoned data augmentations, derived for training classifiers on the chick brain dataset, extended well, classifying another limited microscopy dataset of developing limbs with high accuracy.

Saliency maps identify novel class-specific features

We next performed saliency analysis (16), initially on the brain datasets, to determine to which region(s) of our images our classifiers were sensitive. We selected the baseline model, and a range of models trained on different augmentation regimes (**Table 1**), and first visualised their saliency maps for test images of each brain sub-stage (**Fig 3Ai-vi**, **S4 Fig**

Ai-vi, **S4 Fig Bi-vi**). Our baseline model focused on the brain at 10.2, with scattered attention peripherally. (**Fig 3Ai**). This was reflected in the 10.1 maps (**S4 Fig Ai**), but not the 10.3 maps (**S4 Fig Bi**). Models trained with Gaussian blur (**Fig 3Aii**), cutout (**Fig 3Aiii**), and shear (**Fig 3iv**) honed focus substantially, leaving several focal points (compare **Fig 3Ai** with **Fig 3Aii-iv**). Similar effects were seen in the 10.1 and 10.3 maps (**S4 Fig Aii-iv, S4 Fig Bii-iv**). Gaussian blur also introduced focus on the 10.2 prosencephalic neck (**Fig 3Aii**). We included crop as a negative control, as it showed the worst performance of all augmentations tested (**Table 1**). Crop either produced very little focus (**Fig 3Avi**) or scattered attention throughout the image (**S4 Fig Avi**, **Bvi**). These results confirm that the choice of augmentation results in different image features being picked out by a classifier, but also that biologically relevant features e.g. the prosencephalic neck can be focused on after augmentation with Gaussian blur.

We next tested the DCNN trained on our combined regime, applying either Gaussian blur, cutout, or shear over the baseline per image. This regime produced a similar average validation accuracy and standard deviation to Gaussian blur (**Table 1**, the highest scoring individual augmentation). The saliency maps show large focal points on the centre of the prosencephalic neck (**Fig 3Bi**, magenta arrow) and the junction of the prosencephalon-prosencephalic neck (**Fig 3Bi**, white arrow). We also saw a 'combined map' effect, where the combined regime produced focus resembling larger versions of focal points from the individual augmentation regimes (**Fig 3A**, **S4 Fig Av, Bv**). In particular we found clear differences in the focal points paid to each class. For 10.1, the combined augmentation regime-trained DCNN focused on the prosencephalic neck and the prosencephalon (**S4 Fig Ci**, magenta and white arrow respectively), while for 10.3, focal points were more posterior and confined to the neuromeres (structures delineating future brain regions) (**S4 Fig Cii**, white arrows). Contrastingly, for 10.2 the focus was spread over the prosencephalon and more peripherally (**Fig 3B**), indicating that this sub-stage shares similarities with both 10.1 and 10.3, being between them in developmental time. Exemplifying the utility of our approach, we found that the sub-stage specific features of 10.3 focused on these more posterior regions (compare **S4 Fig Cii** and **Fig 3B, S4 Fig Ci**), which were not explicitly part of our labelling criteria.

In summary, the brain classifier saliency maps confirmed our labelling intuitions (for the 10.1 and 10.2 sub-stages), and provided insight into novel morphological features that best defined the 10.3 sub-stage, outside of our preconceptions.

We then generated saliency maps in a similar fashion for the limb classifier. For the control limb our baseline augmentation regime produced a tight focus on the posterior distal tip (**Fig 4Ai**). Gaussian blur and shear introduced some focus more anteriorly in the limb (**Fig 4Aii, iv**). Cutout produced a similar map to baseline, albeit with slightly reduced focus (**Fig 4Aiii**). Whilst crop performed relatively better than in the brain classification, we again included it for comparison, as it continued to underperform every other augmentation regime. The crop saliency map resembled the baseline, but with a reduced focus (**Fig 4Avi**).

For the treated limb, we saw more attention paid to the anterior proximal edge in the baseline, Gaussian blur, and shear regimes (**Fig 4Bi, ii, iv**). Again this was reduced slightly in the cutout map (**Fig 4Biii**). Shear introduced a focal point more distally, whilst retaining the focal points that the other maps exhibited. Again, crop produced very little focus, with one large point at the corner, likely reflecting the expanded field of view of the test image compared to the cropped augmentation regime.

We next tested our combined regime, augmenting each baseline regime image with Gaussian blur, shear, or cutout. For both the control and treated limbs, we saw a large increase in focus from this regime (**Fig 4Bv**). Intuitively and similar to the brain saliency maps, we saw a combined map effect - where the combined regime introduced focal points from each other augmentation regime. Additionally, we see clearest in the combined regime saliency maps that the classifiers define the two classes based on different aspects of the tissue morphology. For the control limb, we saw this regime focus on the overall width of the limb (**Fig 4Ci** upper white arrow, **S5 Fig Bi**) and the posterior distal edge (**Fig 4Ci,** lower white arrow). For the treated limb, we saw a focus on a more anterior distal edge (**Fig 4Cii**, magenta arrow), as well as a strip of focus running from proximal to distal, suggesting that the width of this axis in the limb was important in classification. Surprisingly, despite the fact that both the control and growth inhibitor-treated limbs had been processed to visualise the expression of the Sonic Hedgehog (SHH) gene (**Fig 4Ci-ii,** orange arrows, **S5 Fig A-Bi**), the classifier did not pay attention to these regions of the image. This is important, as SHH is intricately linked to developing tissue fate and growth in the limb (31) (and other contexts). In

the limb dataset, there were images which have no SHH expression.We also found that SHH presence was not a classifying feature in test cases where SHH was switched off (**S5 Fig A-Bii**)

These results suggest that rigorous examination of classifier attention gives insight not only into data processing and augmentation efficacy, but also into the features of the embryo which most define the classification, with the potential for insights which help examine the relationship between morphology and other biological factors, including gene expression.

DISCUSSION

In this work, our approach was to combine the objective, systematic nature of machine learning approaches with biological domain expertise. This enabled training of high accuracy image classifiers on limited datasets, through bespoke data processing and augmentation regimes.

The key novelty of the study is the development of a strategy that enables classification of limited microscopy datasets covering subtle morphological changes in different experimental systems. We report a novel staging problem in early chick brain development, namely that the early brain undergoes subtle morphological changes over a very short time window, and that this phenomenon is uncoupled from somite number. This agrees with other studies on different developing tissues. For example, asynchronies have been reported in somite number and both mouse limb morphology (21), and frog neural development (22). Therefore, we and others find that the number of somites is uncoupled from wider organ development at certain stages. Due to this uncoupling, traditional staging methods fail. Our trained image classifier provides a solution to classify the brain as it undergoes subtle morphological change.

Development is a continuous process, so a critical question is how to categorise development into discrete tractable stages. The methodology used here provides an unbiased and systematic way of determining the number of classes to separate developmental staging data. In this instance, clustering of the brain dataset suggested three classes, but in other cases the class number may be more/less and our approach naturally extends to that.

Further work could expand the dimensions of the images used. We used 2-D morphological profiles to train our classifiers. Extending this with 3-D fluorescent images, which are increasingly used in developmental biology, could provide a more robust, accurate classifier. Additionally, including gene expression data for the brain classification could couple our sub-stages to a biological mechanism(s). Finally, the classifier could be used more in the developing limb (21), for example following our sub-staging approach, to provide a more fine-grained profile of limb development.

Previous efforts to classify microscopy images in developmental biology have focused on hyperparameter optimisation (7) without data augmentation. In contrast, we have instead focused on exploring different data augmentations, as we can fully exploit biological domain expertise when augmenting data, as opposed to hyperparameter optimisation. Recent theoretical work supports our approach, highlighting the superiority of data augmentation at optimising the learning of DCNNs (32). We saw substantial differences in model performance with different data augmentations, indicating the importance of reasoned augmentation regime design. We also found that combining individually successful augmentations was an effective method to consolidate the benefits of these augmentations. To exemplify the utility of our approach, we then applied similar data augmentation regimes to a developing limb dataset. Again, we made reasoned predictions about the dataset, for instance, that the model would overfit without flipping augmentations. We conclude that careful analysis of the data and an appropriate augmentation regime should be designed for specific datasets and problems encountered in developmental biology.

Our studies indicate that training using InceptionV3 and ResNet50 architectures is not always suitable for all classification problems, specifically on small datasets. This is unsurprising, as these models were built for ImageNet, a more general and larger dataset. Indeed, other work on biological image classification using small (albeit larger than ours) datasets has encountered issues using a ResNet50 architecture. Margapuri et al. tested various model architectures (including ResNet50) at classifying different bee species (33). They found that ResNet50 overfits on their data, achieving at best a 7.4% accuracy. They also tested InceptionV3 finding similar results to ours, with a maximum accuracy of 44.5%. Whilst these authors did test augmentations including rotation and cutout, they found that the un-augmented data achieved higher accuracies. Considering these and our findings, it is

clear that applying widely used image classification architectures alone is not a straightforward solution when dealing with small datasets.

The approach that we take here - i.e. training a DCNN to classify images has advantages over more traditional classifiers which rely on precomputed features of the dataset. This latter method requires arduous manual selection and measurement of image features. In contrast, our approach requires no prior knowledge about the embryos, nor the skill to assess the morphological features. Our approach also shows that pre-selected features may not be those that are the most useful for classification. For instance, we find that the best classifier of the 10.3 brain sub-stage is neuromere shape (not one of the pre-selected features), while for the limb, morphology was a stronger classifying feature than *SHH* expression, even in cases where *SHH* was expressed in control and not in the treated limb. The strategy we present here would also allow image classifiers to be trained for other biological problems with limited microscopy data. Finally, using our DCNN is straightforward for the end user, an important factor when considering the time critical nature of biological experiments.

Advances in higher-resolution data collection methods (34), including detailed time-courses of embryonic development by microscopy, necessitate careful consideration of traditional staging guides. Our studies underscore the importance of developing novel methods to finely gauge development in the early brain: we find that the number of somites is out of sync with brain development. This has consequences for experiments on the development of early brain regions, including the hypothalamus, where accuracy may ultimately be key in treating various disorders, such as obesity and chronic stress (35–38).

Finally, we used saliency maps to gain insight into the regions of the images to which our classifier was most sensitive. Our results illustrate the utility of saliency analysis in interpreting classifiers in a manner that is coherent to experimental biologists. This has been exemplified in other areas at the interface of DL and biomedical research e.g. X-ray images of lung cancer labelled by radiologists (28,39). Here, the authors leveraged saliency methods to understand classifier decision making, an important step if DNN based classifiers are to have a role in medical decision making. The use of such methods will encourage confidence in non-specialists about DCNN-based classifiers. As we show here, there is potential for insight into the most predictive features of labelled classes, free from human

preconceptions. Therefore, these analyses encourage confidence in the classifier, which will increase uptake by the wider developmental biology field.

MATERIALS AND METHODS

Data acquisition

Ground-truth data used for training and validating the image classifier comprised bright-field and phase contrast microscopy images of HH10 (9-12 somite) chick embryos, and included published and unpublished data. Images were acquired using the following microscopes: Olympus BX60, Zeiss AxioImager.Z1, and a Leica dissecting microscope at 4x or 10x magnification. Training data were labelled into three sub-stages, termed 10.1, 10.2 and 10.3, based on our domain expertise, according to the gross morphology of the HH10 brain. Briefly, stages were assigned according to the overall shape of the prosencephalon, the angle of the posterior prosencephalon relative to the prosencephalic neck, and optic vesicle shape. There are 151 images in the dataset, with 54 sub-stage 10.1, 55 sub-stage 10.2, and 42 sub-stage 10.3 embryos respectively. The limb dataset contained 269 images, with 150 labelled as control, and 119 labelled as treated, and were acquired as outlined in (30). The images of both datasets were JPEG format, and varied in resolution, from 188 x 188 to 1000 x 1000 pixels.

Statistical analysis of somite numbers

We did not assume that the variance of somite number with sub-stages is evenly distributed. Therefore, this relationship was analysed by unpaired *t*-test with Welch's correction.

Clustering analysis

We first reduced the dimensionality of the raw images via principal component analysis (PCA) (40). We determined the appropriate number of PCs to be 2 by iteratively increasing this number from 1 until we found diminishing returns in the proportion of variance explained. We then performed *k*-means clustering on the dimensionally-reduced dataset (41), determining the appropriate number for *k* to be 3 by iteratively increasing this number from 1

until we found diminishing returns in the reduction in within-cluster sum of squares (WCSS) (42). All clustering was implemented in Python 3.7.12 using scikit-learn 0.24.1 (43). **Figs 1-2** were generated using seaborn 0.11.2 (44) and matplotlib 3.2.2 (45).

Data preprocessing

We tested a variety of data augmentation regimes (**Tables 1-2 S2-3 Tables, S2 Fig**). To make the trained model invariant to image features that are not classifying (e.g. scale, colour), various preprocessing methods were applied. Images were converted to grayscale, resizing to 200 x 200 pixels using openCV (3.4.2.17) and pillow (8.3.1) (46,47). 200 x 200 is sufficiently small to be easily processed, whilst retaining sufficient spatial resolution to distinguish morphology. We normalised the histograms of each image, using the pillow function ImageOps.equalize (47). This served to brighten images that were too dark and vice versa. For sharpening, (**Table 1**) we used the pillow function ImageFilter.SHARPEN (47).

Data augmentation

Following preprocessing, we constructed a baseline augmented dataset. For microscopy images, a common user-generated feature during data collection is the orientation of the sample. Thus we included rotation as a baseline augmentation, choosing to rotate each image by 36 multiples of 10°. Our rotation method enlarged the images on rotation without cutting off any part. This meant that in addition to rotational and colour invariance, scale invariance would be included into the baseline datasets. For the limb classification, we also incorporated flipped images as part of the baseline, ensuring the left/right symmetry. All augmentations were implemented with imgaug 4.0 (48).

Traditional classifiers

To construct our Random Forest, Support Vector Machine (SVM), and *k*-Nearest Neighbour (KNN) classifiers, we used scikit-learn (43). For each, we fit 10 separate models, generating a new training and validation split for each model. Our splitting followed a 80:20 ratio (120 training, 31 validation images).

Cross-validation

Initially, we used hold out cross-validation, where a portion of the dataset (in our case, 20%) is held back and used as a validation set. We trained 10 models per augmentation regime, reshuffling the training/validation data each time. Our results were substantially variable, e.g. for models trained on our Baseline dataset, the difference between the minimum and maximum validation accuracy was 26.9% (**S4 Table**). We supposed that this variability was a product of our small dataset. As 20% of the dataset was being held back, even just a few skewed features present in the training or validation set would affect results. Therefore, we employed *k*-fold cross validation (**Tables 1-2, S2-3 Tables**). This ensured we made use of the entire dataset, as well as isolating the portions of the data contributing to variable results (49). Briefly, we partitioned the dataset into 10 non-overlapping folds of 15 images. We then trained the network on folds 2-10 and validated with the first fold. Following this, the network was then trained with folds 1 and 3-10, with the second subset used to validate. This proceeded until all folds were used. In this way, we averaged performance of our neural network across the entire dataset.

Neural network architecture

We based our bespoke DNN on the Visual Geometry Group (VGG-16) model architecture, which has been highly successful for image classification (29) (**S3 Fig**). This architecture involves repeated functional units or VGG 'blocks', each comprising a convolution layer with resolution preservation followed by a max-pooling layer that performs two-fold spatial down-samplin. We depart here from VGG-16, as we only include one convolutional layer between each max pooling layer. Between the convolutional and max-pooling layers there is a Rectified Linear Unit (ReLU) activation function (50). As the actual spatial resolution of the data decreases, the number of filters doubles. Thus, the first layer that receives the 200 x 200 image input has 16 functional units, which is repeated 6 more times resulting in a final convolutional layer with dimensions 4 x 4, with 1024 functional units. This follows a similar pattern to VGG-16, however the largest convolutional layer in VGG-16 is 512 wide, whereas we extend to 1024. Following these blocks, we include a fully connected layer of 1024 units, followed by a softmax classification layer for the brain classification, or a sigmoid classification layer for the limb classification. Here, our model differs from VGG-16, which uses three wider fully connected layers. All neural networks were built with Keras 2.3.0 (51) and trained using TensorFlow 2.2.0 (52).

InceptionV3 and ResNet50

We tested two common neural network architectures: InceptionV3 (26) and ResNet50 (27). First, we randomised the weights, before training for 1-50 epochs (full training cycles). The exact number of epochs varied as we implemented early stopping, halting training if 10 epochs had passed without a change in validation accuracy over a 0.01% threshold. The number of epochs to pass, and the early stopping threshold, were selected empirically based on the speed at which models that were allowed to run to 50 epochs converged. When this was triggered, we restored the highest scoring weights in training before saving the model. Following the typical approach (53) we inserted a softmax classification layer as the last layer in the model. The softmax activation performs the actual classification by transforming the input between 0 and 1 outputting three values which sum to 1, which effectively define probabilities of the input belonging to each sub-stage. This was necessary as both InceptionV3 and ResNet50 were designed around the ImageNet dataset (1000 classes). We used the optimiser Adam with an initial learning rate of 10^{-4} (54), but decreased this to 10^{-5} after we observed overfitting. However, 10^{-5} is also commonly used for image classification (33,54). Hyperparameter fine-tuning was implemented using TensorFlow 2.2.0 (52) and Keras 2.3.0 (51).

Training regime

As above, we trained for no more than 50 epochs, and implemented early stopping. We regularised our network with L_2 regularisation (weight decay), which penalises large weights in the neural network (53). The key parameter, λ, is a fraction of the sum of the squared weights of the network. As λ increases, the loss function value increases. Because a neural network is optimised by minimising the loss function, L_2 regularisation encourages smaller weights and thus less complex models. We chose the numerical value of λ through Bayesian optimisation (**S6 Table**) as 10^{-4}. We find that this value has been found to be effective in training image classifiers (55). We also used dropout, which randomly turns off neurons in a layer at a given rate (56). This discourages individual neurons from becoming dominant, encouraging a classifier with better generalisability. We added a 20% dropout layer between each convolutional and max pooling layer, and a 50% dropout layer before the final classification; these percentages were also determined through Bayesian optimization (**S6**

Table). We used the optimiser Adam, with a learning rate of 10^{-4}, determined through Bayesian optimisation (**S6 Table**). We set our range of trialled learning rates to test during optimisation (10^{-1}–10^{-6}) according to our InceptionV3/ResNet50learning rate of 10^{-5}. Our model has fewer trainable parameters than both InceptionV3 and ResNet50. Due to this, our training converges more slowly, and therefore it is unsurprising that our model performs optimally with a faster learning rate than ResNet50 or InceptionV3.

Saliency analysis

We generated saliency maps using tf-keras-vis 0.8.0 (57). In these, image pixels are coloured based on whether they contribute positively (hot colours) or negatively (cold colours) towards the output prediction. This produces a map of the features of an input the network deems most and least important towards a classification. We used the SmoothGrad method, which produces clean saliency maps by adding noise to the input image (58). To understand the decision making of the classifier on real-world data, we generated the saliency maps using test images, which were not involved in model training or validation.

Software

The neural networks were built and trained using Python 3.6. The models were trained on a NVIDIA Tesla V100 GPU using the HPC system provided by the Joint Academic Data Science Endeavour (JADE) II. The code and software dependency list are publicly available at https://github.com/ianbgroves/chick_embryo_DCNN_classifier.

DECLARATIONS

Ethics approval

All experiments used to generate the data were carried out according to the UK Animals (Scientific Procedures) Act 1986. No ethical approval or Home Office licensing was required, as chicks were not incubated beyond E3.5 (hatching is E21). Named Animal Care and Welfare Officers (NACWOs) had oversight of all incubated eggs.

Availability of data and materials

The datasets analysed during the current study are available in the GitHub repository, https://github.com/ianbgroves/developmental_morphology_DCNN_classifier.

Competing interests

We have no competing interests.

Funding

This work was supported by the Wellcome Trust (grant number 212247/Z/18/Z to M.P., grant number 202756/Z/16/Z to M.T.); and the Engineering and Physical Sciences Research Council (PhD studentship to I.G.). For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Authors' contributions

I.G. Project conception, chick brain data generation, data analysis, programming, manuscript writing, manuscript review. J.H. Data analysis, programming. D.F. Data analysis, programming. M.T. Chick limb data generation, manuscript review. B.E. Project conception, manuscript review. M.P. Project conception, data labelling, manuscript review. A.G.F. Project conception, manuscript writing, manuscript review. All authors read and approved the final manuscript.

Acknowledgments

We thank S. Burbridge and K. Chinnaiya for providing unpublished images of chick embryos and S. Burbridge, K. Chinnaiya, E. Manning and E. Place for comments on the manuscript.

REFERENCES

1. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44.
2. Iizuka O, Kanavati F, Kato K, Rambeau M, Arihiro K, Tsuneki M. Deep learning models

for histopathological classification of gastric and colonic epithelial tumours. Sci Rep. 2020;10:1–11.

3. Caicedo JC, Roth J, Goodman A, Becker T, Karhohs KW, Broisin M, et al. Evaluation of Deep Learning Strategies for Nucleus Segmentation in Fluorescence Images. Cytometry A. 2019;95(9):952–65.
4. Thompson NC, Greenewald K, Lee K, Manso GF. The computational limits of deep learning. ArXiv Prepr ArXiv200705558. 2020;
5. Jacquemet G. Deep learning to analyse microscopy images. Biochem. 2021;43:60–4.
6. Hallou A, Yevick HG, Dumitrascu B, Uhlmann V. Deep learning for bioimage analysis in developmental biology. Development. 2021;148(18):dev199616.
7. Pond AJR, Hwang S, Verd B, Steventon B. A deep learning approach for staging embryonic tissue isolates with small data. PLoS One. 2021;16:e0244151.
8. Rosin PL, Fierens F. Improving neural network generalisation. In: International Geoscience and Remote Sensing Symposium. IEEE; 1995. p. 1255–7.
9. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE; 2009. p. 248–55.
10. Simard PY, Steinkraus D, Platt JC. Best practices for convolutional neural networks applied to visual document analysis. In: Icdar. 2003.
11. DeVries T, Taylor GW. Improved regularization of convolutional neural networks with cutout. ArXiv Prepr ArXiv170804552. 2017;1–8.
12. Zhou S, Zhang J, Jiang H, Lundh T, Ng AY. Data augmentation with Mobius transformations. Mach Learn Sci Technol. 2021;2:025016.
13. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
14. Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res. 2012;13(2).
15. Shin S, Lee Y, Kim M, Park J, Lee S, Min K. Deep neural network model with Bayesian hyperparameter optimization for prediction of NOx at transient conditions in a diesel engine. Eng Appl Artif Intell. 2020;94:103761.
16. Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ArXiv13126034 Cs. 2014
17. Placzek M, Briscoe J. Sonic hedgehog in vertebrate neural tube development. Int J Dev Biol. 2018 ;62(1-2–3):225–34.

18. Hamburger V, Hamilton HL. A series of normal stages in the development of the chick embryo. J Morphol. 1951;88:49–92.
19. Palmeirim I, Henrique D, Ish-Horowicz D, Pourquié O. Avian hairy gene expression identifies a molecular clock linked to vertebrate segmentation and somitogenesis. Cell. 1997;91:639–48.
20. Boehm B, Rautschka M, Quintana L, Raspopovic J, Jan Ž, Sharpe J. A landmark-free morphometric staging system for the mouse limb bud. Development. 2011;138:1227–34.
21. Musy M, Flaherty K, Raspopovic A J and Robert-Moreno, Richtsmeier JT, Sharpe J. A quantitative method for staging mouse embryos based on limb morphometry. Development. 2018;145:dev154856.
22. Sáenz-Ponce N, Mitgutsch C, del Pino EM. Variation in the schedules of somite and neural development in frogs. Proc Natl Acad Sci USA. 2012;109:20503–7.
23. Newgreen DF, Erickson CA. The migration of neural crest cells. Int Rev Cyt. 1986;103:89–145.
24. Krueger R, Beyer J, Jang W-D, Kim NW, Sokolov A, Sorger PK, et al. Facetto: Combining unsupervised and supervised learning for hierarchical phenotype analysis in multi-channel image data. IEEE Trans Vis Comput Graph. 2019;26(1):227–37.
25. Ben-Hur A, Guyon I. Detecting Stable Clusters Using Principal Component Analysis. In: Brownstein MJ, Khodursky AB, editors. Functional Genomics: Methods and Protocols [Internet]. Totowa, NJ: Humana Press; 2003. p. 159–82. (Methods in Molecular Biology).
26. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2015. p. 1–9.
27. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2016. p. 770–8.
28. Baltruschat IM, Nickisch H, Grass M, Knopp T, Saalbach A. Comparison of deep learning approaches for multi-label chest X-ray classification. Sci Rep. 2019;9(1):1–10.
29. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. ArXiv Prepr ArXiv14091556. 2014;
30. Towers M, Mahood R, Yin Y, Tickle C. Integration of growth and specification in chick wing digit-patterning. Nature. 2008;452(7189):882–6.

31. Groves I, Placzek M, Fletcher AG. Of mitogens and morphogens: modelling Sonic Hedgehog mechanisms in vertebrate development. Philos Trans R Soc B Biol Sci. 2020;375(1809):20190660.
32. Hernández-García A, König P. Data augmentation instead of explicit regularization. ArXiv Prepr ArXiv180603852. 2018;
33. Margapuri V, Lavezzi G, Stewart R, Wagner D. Bombus Species Image Classification. ArXiv Prepr ArXiv200611374. 2020;
34. Cutrale F, Fraser SE, Trinh LA. Imaging, visualization, and computation in developmental biology. Annu Rev Biomed Data Sci. 2019;2:223–51.
35. Bedont JL, Newman EA, Blackshaw S. Patterning, specification, and differentiation in the developing hypothalamus. WIREs Dev Biol. 2015;4(5):445–68.
36. Biran J, Tahor M, Wircer E, Levkowitz G. Role of developmental factors in hypothalamic function. Front Neuroanat. 2015;9:47.
37. Eachus H, Bright C, Cunliffe VT, Placzek M, Wood JD, Watt PJ. Disrupted-in-Schizophrenia-1 is essential for normal hypothalamic-pituitary-interrenal (HPI) axis function. Hum Mol Genet. 2017;26(11):1992–2005.
38. Moir L, Bochukova EG, Dumbell R, Banks G, Bains RS, Nolan PM, et al. Disruption of the homeodomain transcription factor orthopedia homeobox (Otp) is associated with obesity and anxiety. Mol Metab. 2017;6(11):1419–28.
39. Panwar H, Gupta PK, Siddiqui MK, Morales-Menendez R, Bhardwaj P, Singh V. A deep learning and grad-CAM based color visualization approach for fast detection of COVID-19 cases using chest X-ray and CT-Scan images. Chaos Solitons Fractals. 2020;140:110190.
40. Partridge M, Calvo RA. Fast dimensionality reduction and simple PCA. Intell Data Anal. 1998;2(3):203–14.
41. Ranjan S, Nayak DR, Kumar KS, Dash R, Majhi B. Hyperspectral image classification: A k-means clustering based approach. In: 4th International Conference on Advanced Computing and Communication Systems. 2017. p. 1–7.
42. Bholowalia P, Kumar A. EBK-means: A clustering technique based on elbow method and k-means in WSN. Int J Comput Appl. 2014;105(9).
43. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–30.
44. Waskom M, Gelbart M, Botvinnik O, Ostblom J, Hobson P, Lukauskas S, et al. mwaskom/seaborn: v0.11.2. Available from: https://zenodo.org/record/5205191

45. Hunter JD. Matplotlib: A 2D graphics environment. Comput Sci Eng. 2007;9:90–5.
46. Bradski G. The OpenCV Library. Dr Dobbs J Softw Tools. 2000;
47. Clark A. Pillow (PIL Fork) Documentation. readthedocs; 2015. Available from: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf
48. Jung AB, Wada K, Crall J, Tanaka S, Graving J, Reinders C, et al. imgaug. 2020. Available from: https://github.com/aleju/imgaug
49. Fushiki T. Estimation of prediction error by using K-fold cross-validation. Stat Comput. 2011;21(2):137–46.
50. Agarap AF. Deep learning using rectified linear units (ReLU). ArXiv Prepr ArXiv180308375. 2018;
51. Chollet F, others. Keras [Internet]. 2015. Available from: https://keras.io
52. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Internet]. 2015. Available from: https://www.tensorflow.org/
53. Goodfellow I, Bengio A Y and Courville. Deep Learning. MIT Press; 2016.
54. Zhang J, Mitliagkas I. YellowFin and the Art of Momentum Tuning. Proc Mach Learn Syst. 2019;1:289–308.
55. Gabas A, Corona E, Alenyà G, Torras C. Robot-aided cloth classification using depth information and CNNs. In: International Conference on Articulated Motion and Deformable Objects. Springer; 2016. p. 16–23.
56. Srivastava N. Improving neural networks with dropout. Univ Tor. 2013;182(566):7.
57. Yasuhiro K. tf-keras-vis. GitHub repository. GitHub; 2021. Available from: https://github.com/keisen/tf-keras-vis
58. Smilkov D, Thorat N, Kim B, Viégas F, Wattenberg M. Smoothgrad: removing noise by adding noise. ArXiv Prepr ArXiv170603825. 2017;
59. LeNail A. NN-SVG: publication-ready neural network architecture schematics. J Open Source Softw. 2019;4(33):747.

FIGURES

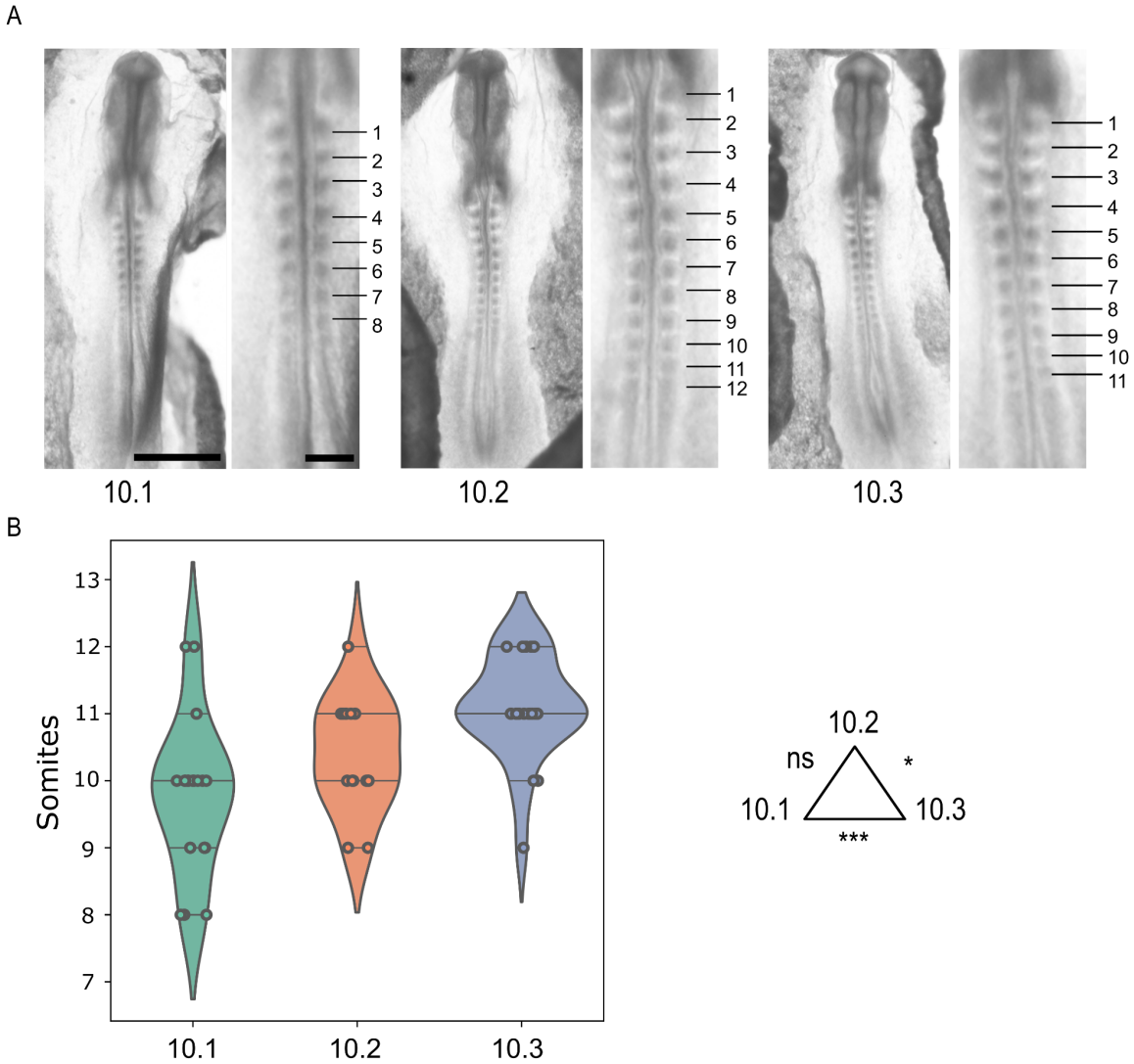


Fig 1. The number of somites at the 10.1, 10.2, and 10.3 sub-stages does not account for the variation seen in HH10 brain morphology. (A) Manual quantification of somite number for the reference images shown in **Fig 1A**. Right hand panels show close-ups of somitic regions. Scale bars: left 1000µm, right 250µm. (B) Violin plot of somite count across sub-stages 10.1, 10.2 and 10.3. (n= 18 10.1, 14 10.2, and 21 10.3 embryos respectively: note the data used for these quantifications were not included as training data). There are statistically significant differences in somite numbers between 10.1 and 10.3, and 10.2 and 10.3, but not 10.1 and 10.2. Despite these differences, each sub-stage can have 9, 10, 11, or 12 somites.

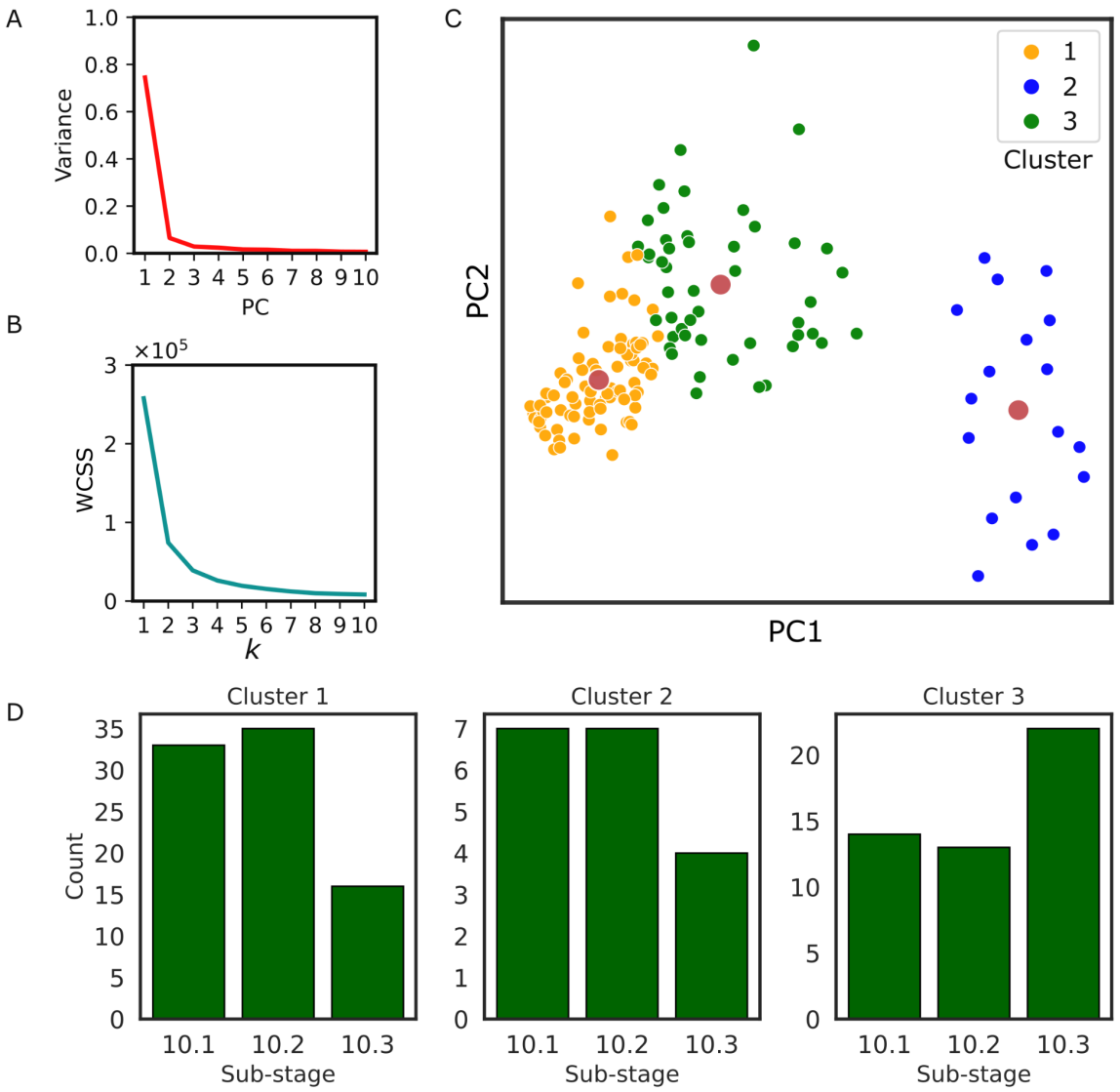


Fig 2. *k*-means clustering supports classification of the image dataset into three sub-stages. (A) Scree plot for principal component analysis (PCA) with increasing numbers of PCs. The elbow point is at 2 PCs, which explain 81% of the variance in the dataset. (B) The elbow method for determining the optimal number for *k*. The within-cluster sum of squares (WCSS) score is calculated for *k*-means clustering on increasing values of *k*. The optimal number of *k* is determined by the inflection point of the graph, which occurs around *k* = 3. (C) Scatter plot of the *k*-means clustered dataset, with *k* centroids (red circles). (D) Number of sub-stages present in each cluster. Despite unsupervised methods identifying three groups of images, the number of embryos in each cluster does not match the number of embryos in each labelled sub-stage of the training data.

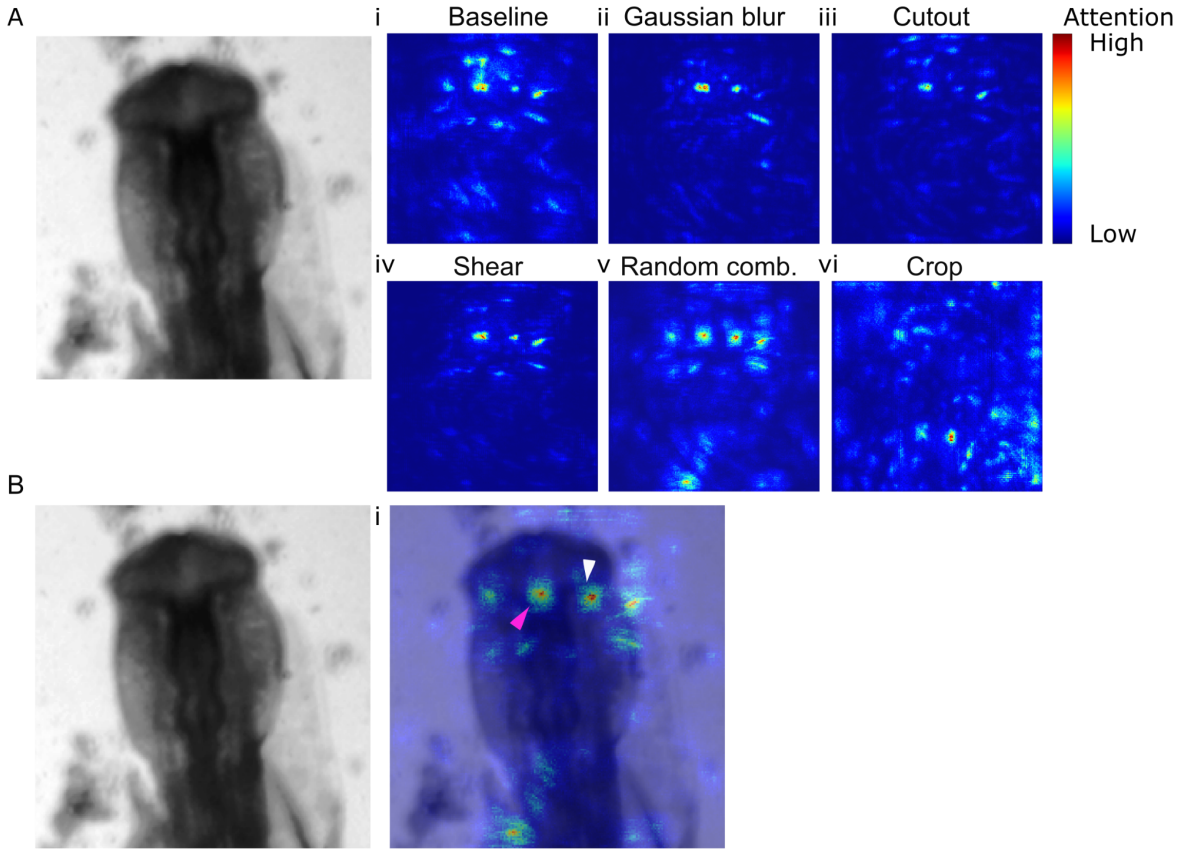


Fig 3. Saliency maps of a 10.2 substage identify focal differences between models trained on different augmentations. (A) A test 10.2 image not used in training/validation of any DCNN. (i-vi) Saliency maps generated from training a DCNN on the baseline rotationally augmented data (i), with a further augmentation (ii-vi, listed above each image). Baseline (i) leads to a large amount of attention spread over the prosencephalon. Gaussian blur (ii) cutout (iii), and shear (iv) increase focus to a few focal points on the prosencephalon. The maps generated by the augmentations i-v contrast with crop (vi). This performed the worst of all our augmentation regimes, and has very little focus on the prosencephalon. (B) (i) The saliency map generated by the combined regime as shown in (Av), overlaid with the test 10.2 image. The combined regime increases the focus on points introduced by the various augmentations, and the focus is highest on the region of the embryo (i, magenta arrow) central to the prosencephalic neck (white arrow), also a region of high focus. The combined regimes of various augmentations increase the focal attention. Highest focal points are the centre of the prosencephalic neck (magenta arrow) and the junction of the prosencephalon-prosencephalic neck (white arrow).

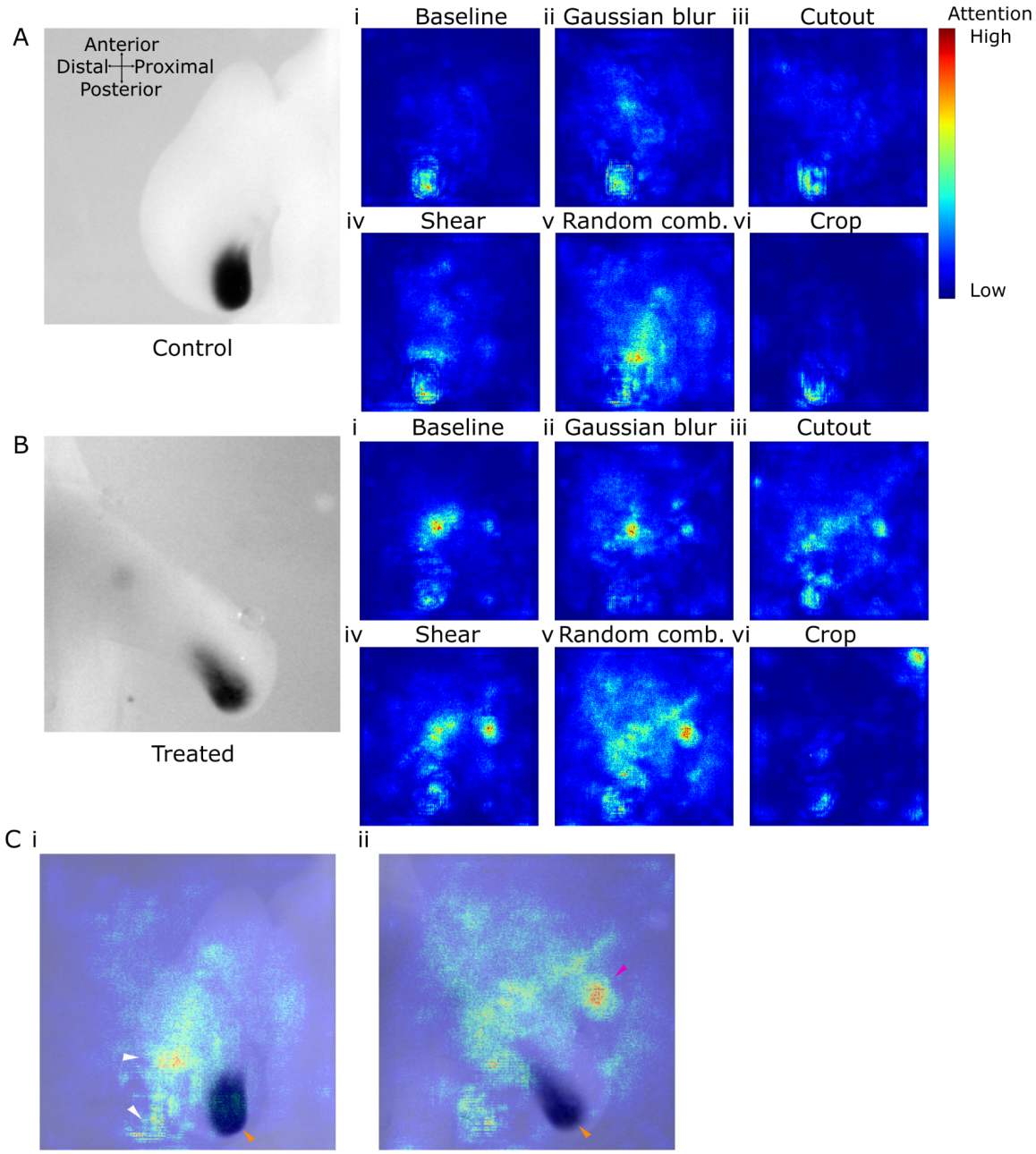


Fig 4. Saliency maps from images of developing chick limbs identify morphology as a class-defining feature. (A) A test control limb image not used in training/validation of any DCNN. (i-vi) Saliency maps generated from training a DCNN on the baseline of flipped and rotationally augmented limb data (i), with a further augmentation (ii-vi, listed above each image). Baseline (i) leads to a tight focal point on the posterior distal edge of the limb. Gaussian blur (ii) cutout (iii), and shear (iv) share this focal point, but for gaussian blur and shear there are focal points added more anteriorly and toward the centre of the developing limb. The maps generated by the augmentations i-v contrast with crop (vi) which performed

the worst of all our augmentation regimes for the brain classification, and has reduced focus from baseline. (B) (i-vi) A test image with augmentations as in (A) but with a treated limb. Baseline (i) exhibits a large focal point in the proximal anterior portion of the limb, with a smaller, posterior distal point. Gaussian blur (ii) resembles baseline, but with smaller additional points of focus introduced throughout the limb. Cutout, and Shear (iii, iv) produce similar saliency maps, with attention paid to both posterior and anterior proximal regions, as well as some posterior distal regions. Again, crop (vi) contrasts with the other saliency maps, in that focus is substantially reduced over Baseline alone. Finally, the combined regime for both control and treated limbs (C) increases focus throughout the limb. For control, the focus is throughout the posterior distal tip, with the highest attention paid to the edges of the limb (C i, white arrows). For the treated limb, we see that focus is generally more anterior and proximal, with the highest focal point being the distal long edge of the developing limb (C ii, magenta arrow).

TABLES

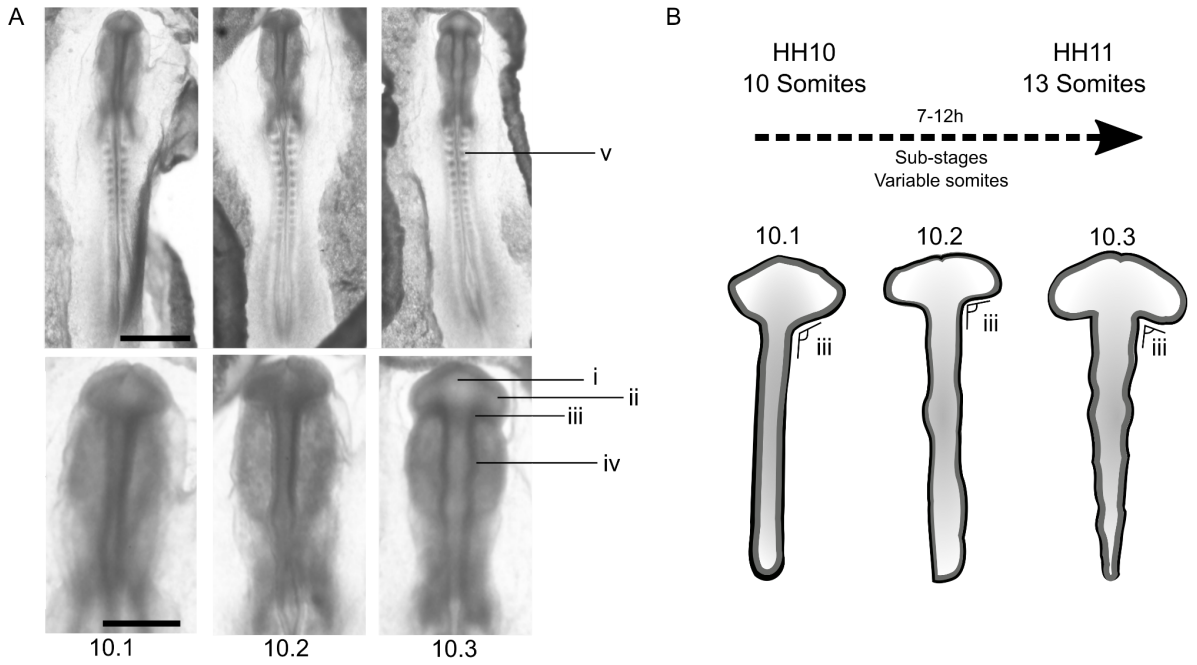
Aug	Fold											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
1	69.3	68.1	79.3	78.9	72.6	84.8	74.6	68.0	90.9	75.2	76.2	7.5
1 Sharpened	78.7	80.9	77.0	70.2	80.2	80.7	70.7	76.7	90.0	76.9	78.2	5.6
1 + 2	68.9	75.5	78.8	79.8	75.1	84.5	78.8	69.1	90.3	83.1	78.4	6.7
1 + 3	75.7	67.1	58.2	64.2	66.2	67.4	68.6	36.6	79.6	68.0	65.2	**11.6**
1 + 4	81.0	73.8	80.6	**79.1**	**81.7**	**86.9**	**79.0**	71.9	88.9	81.9	**80.5**	**5.1***
1 + 5	73.6	76.6	79.9	76.7	76.9	86.1	78.0	68.1	**91.7**	83.3	79.1	6.6
1 + 4,5 RC	**86.3**	**81.0**	78.2	74.2	77.6	82.2	73.2	**77.6**	89.0	83.2	80.3	**5.1***
1 + 2,4,5 RC	84.6	75.5	**82.1**	70.1	81.2	83.1	75.9	77.1	90.9	**84.2**	**80.5**	5.9
Fold Avg.	70.5	69.2	74.7	73.7	72.3	79.5	71.8	66.7	83.2	74.6		

Table 1. Augmentation exploration of the brain dataset using the bespoke model. We explored the data-space using *k*-fold cross validation, the individual fold validation accuracies that each network achieved are shown in columns 1-10, and the averages and standard deviation of these accuracies is shown in the rightmost columns. As a baseline processing step, all images were rotated 15 times, at equally spaced degrees. We then tested augmentations on top of this baseline, before a final test whereby each image was randomly augmented. Augmentations (Aug) as follows: (1) rotation (baseline); (2) shear; (3) crop; (4) Gaussian blur; (5) cutout; (RC) random combination of rotation + cutout, or shear, or blur. Highest validation accuracies for each fold, highest average for each augmentation (Avg.), and lowest standard deviation (SD) are shown in bold. (1 Sharpened), baseline with a preprocessing sharpen filter; (1+ 4,5 RC) random combination of baseline, Gaussian blur, and cutout.

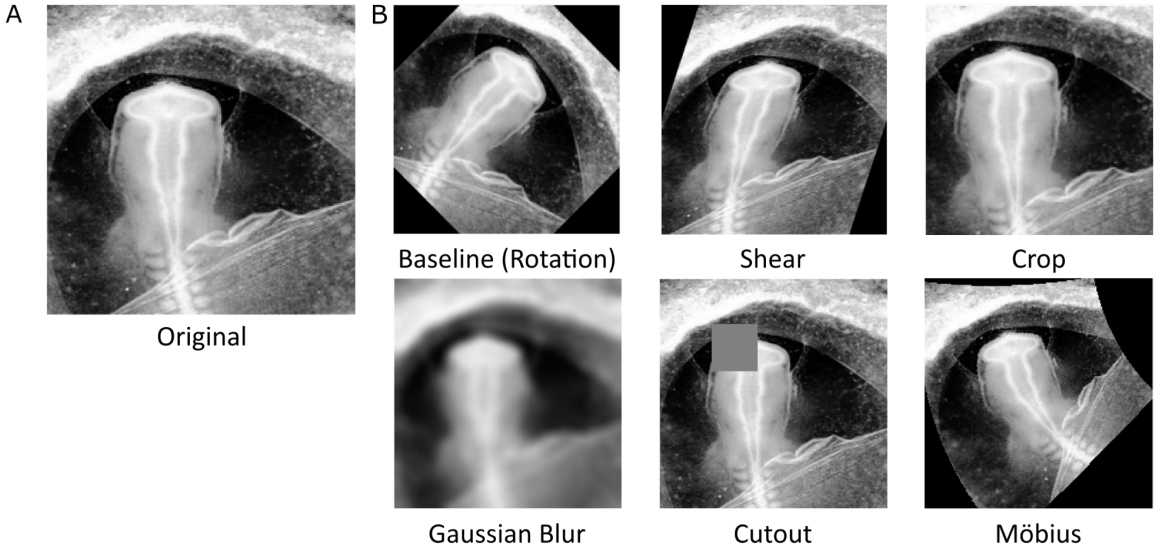
Aug	Fold											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
1 (Flipped)	89.5	**87.0**	83.1	88.0	**91.0**	90.0	91.3	**94.4**	92.9	89.2	89.6	3.2
1 + 2	93.1	80.7	**88.2**	**91.1**	90.1	81.5	**93.6**	92.9	**94.4**	88.9	89.5	4.9
1 + 3	89.1	77.4	83.2	85.9	80.1	81.2	85.2	86.1	87.3	86.9	84.2	3.7
1 + 4	91.1	85.3	82.2	90.1	90.4	**93.6**	90.5	93.0	92.4	90.2	**89.9**	3.5
1 + 5	**93.2**	81.4	88.1	88.6	90.1	85.1	90.7	93.2	93.6	89.8	89.4	3.9
1 + 2,4,5 RC	92.0	84.5	87.5	89.4	90.4	85.5	91.4	93.1	93.6	**91.4**	**89.9**	**3.1**
Fold Avg.	91.3	83.7	85.4	88.9	88.7	86.2	90.5	92.1	92.4	89.4	88.7	

Table 2. Augmentation exploration of a limited, developing limb dataset using the bespoke model. We explored the data-space using *k*-fold cross validation, the individual fold validation accuracies that each network achieved are shown in columns 1-10, and the averages and standard deviation of these accuracies is shown in the rightmost columns. As a baseline processing step, all images were rotated 15 times, at equally spaced degrees and flipped. We then tested augmentations on top of this baseline, before a final test whereby each image was randomly augmented. Augmentations (Aug) as follows: (1) rotation & flip (baseline); (2) shear; (3) crop; (4) Gaussian blur; (5) cutout; (RC) random combination of rotation + cutout, or shear, or blur. Highest validation accuracies for each fold, highest average for each augmentation (Avg.), and lowest standard deviation (SD) are shown in bold.

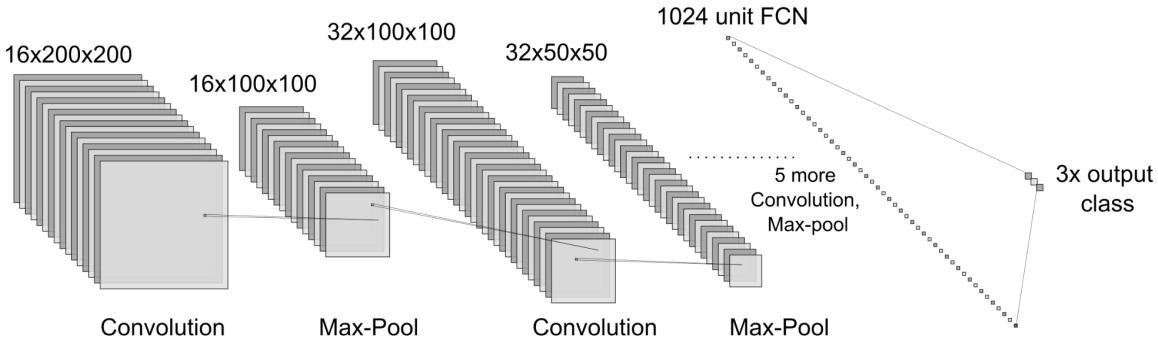
SUPPLEMENTARY FIGURES



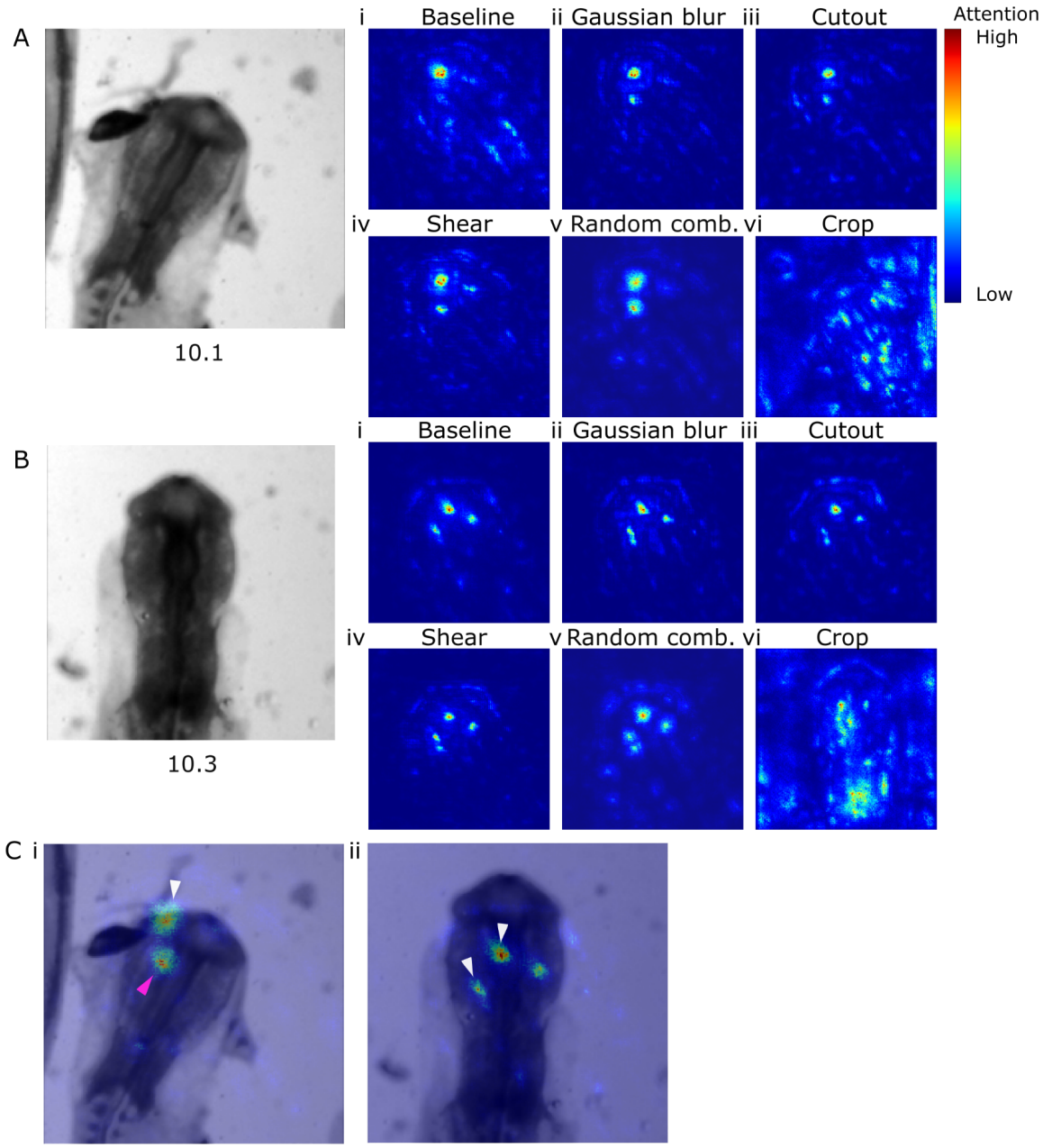
S1 Fig. Sub-staging the HH10 chick embryo brain based on expert visual inspection of morphology. (A) Example brightfield microscopy images of stage 10 sub-stages (top panels), with close-ups of the developing brain (bottom panels). These images were used as the original reference point for ground truth labelling, and were not used as part of the DNN classifier training dataset. (i-iii) show key anatomical markers used for sub-stage labelling. (i): prosencephalon, (ii): optic vesicle, (iii): angle of prosencephalon and prosencephalic neck. (iv, v): show other anatomical features relevant to this study: (iv) somites; (v) neuromere. Scale bars: top 1000μm, bottom 500μm. (B) Schematic indicating morphological profiles of sub-stages.



S2 Fig. Applying different augmentations to an image of a 10.2 sub-stage embryo. (A) An example image of a 10.2 embryo from the dataset (base), with (B) the augmentation regimes tested for classifier training: geometric (e.g. rotation), photometric (e.g. contrast), and complex (e.g. cutout) augmentations.

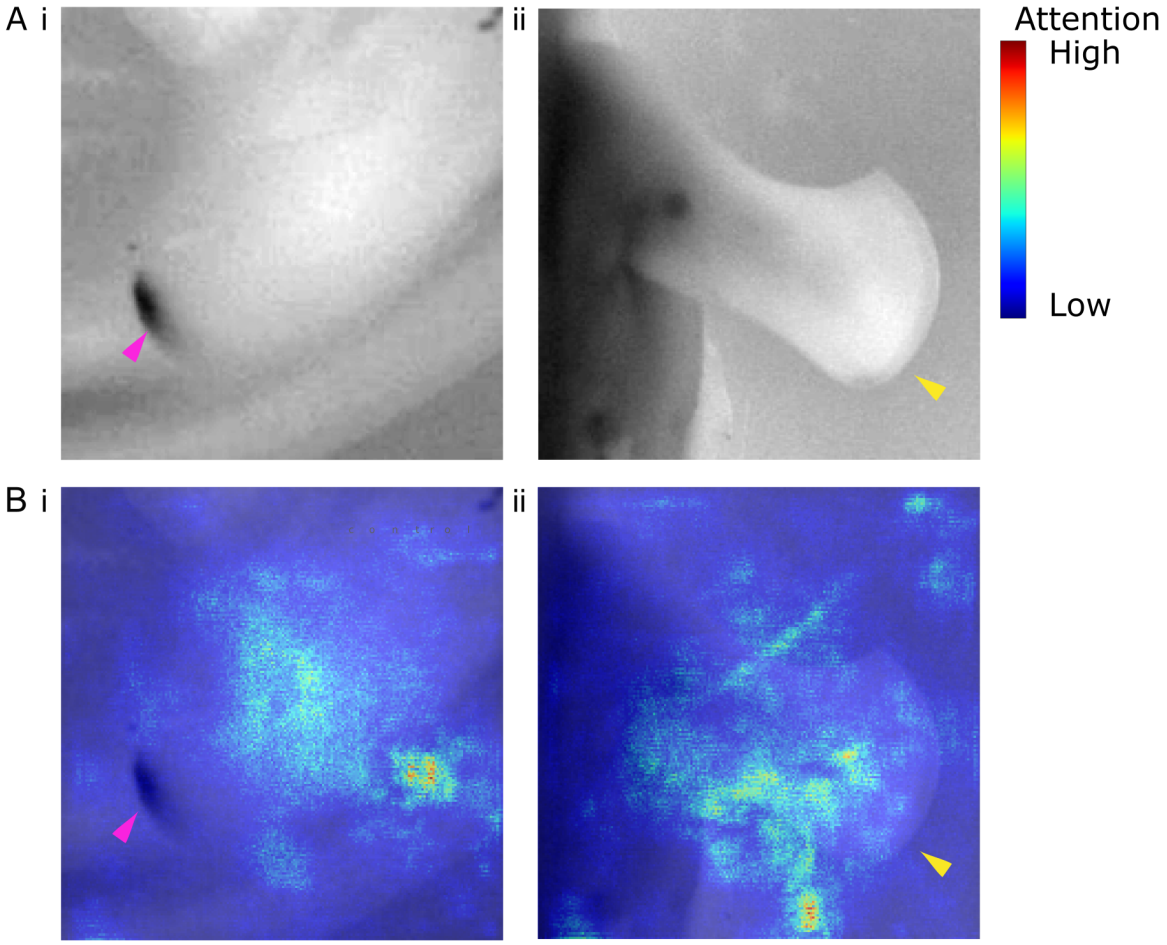


S3 Fig. Image classifier network architecture. After the input layer, the image was processed following the VGG block style, i.e. through increasingly wide convolution and max pooling layer pairs (29). At each 'block' we used a single convolutional layer which retains resolution through padding. Following this, a max pooling layer is used as a method of spatial downsampling. Between each convolutional layer and max pooling layer there is a non-linear ReLU activation function and a dropout layer. For simplicity, only 2 of the blocks are shown. After the final max pooling layer, the data was passed through a wide fully connected layer. Lastly, the classification is made in the output layer. Diagram generated using NN-SVG (59).



S4 Fig. Saliency maps of 10.1 and 10.3 substages identify focal differences between models trained on different augmentations. (A-B) Test images of 10.1 and 10.3 embryos. (i-vi) Saliency maps generated from (A-B) and assessed through models trained on different augmentation regimes, following our architecture (from **Table 1**), trained on the baseline rotationally augmented data, with a further augmentation (listed above each image). Random comb. refers to a model trained on our baseline and a random combination of Gaussian blur, cutout, and shear augmentations. The random combination results in attention which is discrete and focused e.g. (Av), (Bv). Conversely, models trained on

baseline & crop show scattered attention, where the attention map shows high attention pixels scattered throughout the image can be seen (Avi,Bvi). A scattered attention map suggests that the model is focusing on various image features unrelated to the subject. (C) Saliency maps as in (A,B v) overlaid with the test image. For 10.1, the model focused most on the angle of the prosencephalic neck (Ci magenta arrow) and the anterior prosencephalon (Ci white arrow). For 10.3 the focus was shifted to more posterior regions of the embryo, including the neuromeres (Cii, white arrows).



S5 Fig. SHH expression is not used to classify between control and treated limbs. (A) Test images of control (i) and treated (ii) chick embryo limbs, not used in the model training/validation pipeline. The control limb has expression of SHH (i, magenta arrow), whilst the treated limb does not (ii, yellow arrow). (B) Saliency maps generated using the test images in (A), classified using the same mixed augmentation model used to generate Fig 4C i-ii. The network pays little attention to the SHH expression (i, magenta and ii, yellow arrows), instead focusing on the broader shape of the limbs and specific points of curvature.

SUPPLEMENTARY TABLES

	Repeat											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
RFC	**54.8**	25.8	35.5	25.9	32.3	41.9	35.5	29.0	38.7	45.2	36.5	9.1
SVM	22.6	**41.9**	32.3	29.0	22.6	38.7	35.5	22.6	32.3	35.5	31.3	**7.0**
KNN	41.9	38.7	**51.2**	45.2	41.9	32.3	38.7	29.0	22.6	41.9	**38.3**	8.3

S1 Table. Traditional machine learning classification on our un-augmented dataset. Our original dataset was used to fit ten classifiers, and the classification accuracies were determined with a different (80:20) split of training/ testing data for each model. RFC; Random forest classifier. SVM; support vector machine, KNN; *k*-nearest neighbours (k = 3). Highest classification accuracies for each repeat, highest average for each classifier (Avg.), and lowest standard deviation (SD) are shown in bold

Aug	Fold											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
1	40.7	41.5	43.3	46.7	42.4	46.1	47.0	46.7	46.5	40.4	44.1	**2.7**
1 + 2	**41.0**	40.0	53.3	49.1	**44.9**	38.1	46.8	46.9	47.5	39.4	44.7	4.9
1 + 3	35.6	**50.7**	51.3	**49.6**	43.3	47.6	46.2	**47.1**	32.9	**49.1**	45.3	6.3
1 + 4	40.0	35.4	54.4	40.9	40.9	46.6	46.7	46.7	**48.7**	42.6	44.3	5.4
1 + 5	40.1	41.9	53.8	45.4	36.2	47.5	46.7	46.7	46.8	40.1	44.5	5.0
1 + 2,4,5 RC	38.0	42.2	**60.5**	38.7	40.9	**52.8**	**48.6**	46.0	47.1	37.8	**45.4**	7.4
Fold Avg.	39.2	42.0	52.8	45.1	41.4	46.5	47.0	46.7	45.1	41.6		

S2 Table. Augmentation exploration of the dataset using InceptionV3. We explored the data-space using *k*-fold cross validation, the individual fold validation accuracies that each network achieved are shown in columns 1-10, and the averages and standard deviation of these accuracies is shown in the rightmost columns. As a baseline processing step, all images were rotated 15 times, at equally spaced degrees. We then tested augmentations on top of this baseline, before a final test whereby each image was randomly augmented. Augmentations (Aug) as follows: (1) rotation (baseline); (2) shear; (3) crop; (4) Gaussian blur; (5) cutout; (RC) random combination of rotation + cutout, or shear, or blur. Highest validation accuracies for each fold, highest average for each augmentation (Avg.), and lowest standard deviation (SD) are shown in bold.

Aug	Fold											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
1	40.0	71.1	73.5	54.1	69.4	66.9	64.1	70.9	65.9	63.7	64.0	10.0
1 + 2	65.4	40.1	67.0	51.0	69.8	67.1	63.4	48.7	67.9	56.2	59.9	11.7
1 + 3	44.2	55.6	60.9	52.4	51.8	46.7	46.7	50.0	45.8	46.9	50.1	**5.2**
1 + 4	**69.0**	**74.5**	**75.9**	51.1	**70.6**	69.2	**70.6**	70.1	**71.6**	**69.4**	**69.2**	6.8
1 + 5	45.7	40.1	56.9	52.0	44.3	**70.4**	69.8	49.9	71.3	48.2	54.9	11.7
1 + 2,4,5 RC	63.1	37.2	68.7	48.1	67.3	62.7	52.0	**71.0**	68.4	55.3	59.4	11.0
Fold Avg.	54.6	53.1	67.2	51.5	62.2	63.8	61.1	60.1	65.2	56.6		

S3 Table. Augmentation exploration of the dataset using ResNet50. We explored the data-space using *k*-fold cross validation, the individual fold validation accuracies that each network achieved are shown in columns 1-10, and the averages and standard deviation of these accuracies is shown in the rightmost columns. As a baseline processing step, all images were rotated 15 times, at equally spaced degrees. We then tested augmentations on top of this baseline, before a final test whereby each image was randomly augmented (see **Table 1**).

Aug	Avg. (%)	SD	Min (%)	Max (%)
Rotation	80.3	6.7	64.7	**91.6**
Crop	74.8	8.4	59.9	82.3
Shear	78.4	4.8	68.1	84.8
Gaussian blur	80.3	4.9	71.9	89.3
Cutout	**83.6**	**4.6**	**76.1**	91.5
Crop	53.5	8.9	39.2	74.9

S4 Table. Data augmentation regime exploration with hold-out cross-validation. The networks were initialised and trained with reshuffled data 10 times for each augmentation. Rotation achieves the highest accuracy at 91.6%, however cutout reaches 91.5%, and is the most reliable, with the highest average and lowest standard deviation. The optimal score for each metric is shown in bold.

Aug	Fold											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Avg.**	**SD**
1 + Möbius	55.7	57.6	47.4	49.1	49.4	41.3	38.1	57.4	43.9	66.1	50.6	8.6
1 + M:G (10% chance)	76.6	79.4	67.4	66.4	79.8	76.9	70.8	77.6	84.6	79.0	75.9	5.8

S5 Table. Testing of Möbius transformations as data augmentations for the brain dataset. 1 + Möbius: the dataset is augmented with our baseline & Möbius transformations. 1 + M:G (10% chance). The dataset is augmented with Gaussian blur with a 10% chance of a Möbius transformation per image.

Hyperparameter	Value/category
Activation function	ReLU
Batch size	32
Layer dropout	20%
Final layer dropout	50%
L_2 regularisation λ	10^{-4}
Optimiser	Adam
Learning rate	10^{-4}
Average validation accuracy	64.1%
Min, Max validation accuracy	43.6%, 81.5%

S6 Table. Optimal hyperparameters for our baseline, determined by Bayesian optimisation. We tested these hyperparameters with the following ranges: Activation function: ReLU-Sigmoid, Batch size: 16-128, Optimiser: Adam, Adadelta, Adamax, Adagrad, SGD, RMSprop, Layer dropout: 0.05-0.25%. Final layer dropout: 0.3-0.8%. λ: 10^{-3} - 10^{-6}, Learning rate: 10^{-1} - 10^{-6}, selecting the value/category which was used most by the optimisation algorithm.