X-CNN: Cross-modal Convolutional Neural Networks for Sparse Datasets

Petar Veličković[*‡], Duo Wang[*], Nicholas D. Lane[†‡] and Pietro Liò[*]
[*]Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK
[†]Department of Computer Science, University College London, London WC1E 6BT, UK
[‡]Nokia Bell Labs, Cambridge CB3 0FA, UK
Email: {pv273, wd263, pl219}@cam.ac.uk, niclane@acm.org

Abstract—In this paper we propose *cross-modal convolutional neural networks* (X-CNNs), a novel biologically inspired type of CNN architectures, treating gradient descent-specialised CNNs as individual units of processing in a larger-scale network topology, while allowing for unconstrained *information flow* and/or *weight sharing* between analogous hidden layers of the network—thus generalising the already well-established concept of neural network *ensembles* (where information typically may flow only between the output layers of the individual networks). The constituent networks are individually designed to learn the output function on their own subset of the input data, after which cross-connections between them are introduced after each pooling operation to periodically allow for information exchange between them. This injection of knowledge into a model (by prior partition of the input data through domain knowledge or unsupervised methods) is expected to yield greatest returns in sparse data environments, which are typically less suitable for training CNNs. For evaluation purposes, we have compared a standard four-layer CNN as well as a sophisticated FitNet4 architecture against their cross-modal variants on the CIFAR-10 and CIFAR-100 datasets with differing percentages of the training data being removed, and find that at lower levels of data availability, the X-CNNs significantly outperform their baselines (typically providing a 2–6% benefit, depending on the dataset size and whether data augmentation is used), while still maintaining an edge on *all* of the full dataset tests.

I. Introduction

In recent years, the number of success stories of machine learning has seen an all-time rise across a wide range of fields and tasks, examples including: computer vision [1], speech recognition [2], reinforcement learning [3] and guiding Monte Carlo tree search [4]. The unifying idea behind all of the above is *deep learning*, the utilisation of neural networks with many hidden layers, for the purposes of learning complex feature representations from raw data, rather than relying on hand-crafted feature extraction.

As the networks become deeper, however, they become more and more reliant on the amount of training examples provided for maximising their performance. While we are now able to extract large quantities of labelled information for many problems of interest, there remains a significant proportion of tasks for which "big data" simply isn't available at this time, which makes it extremely difficult to fully exploit a deep CNN architecture and properly learn generalisable features of the

data. Here we will present an architectural methodology that attempts to extract additional predictive power from a convolutional neural network (CNN) in such circumstances by instead focussing on the *width* of the data, i.e. the heterogeneity of information present within each training example. The key idea constitutes appropriate *partitioning* of this information and training *smaller CNNs* on these partitions (allowing them to train faster and more effectively under sparse data environments), while allowing for information exchange between them at various stages (Fig. 1).

A classic example where such an approach is bound to be useful are *clinical studies*, where there typically may not be that many patients, but for each patient there is potentially a heterogeneous wealth of information, such as various test results, patient history, ethnic background, body scans (CT, MRI...) and so on, depending on the type of study.

II. Cross-modal CNNs

Our methodology is inspired by *multilayer networks* [5], mathematical structures encompassing several layers of graphs over the same set of nodes, allowing for unrestricted intra-layer as well as inter-layer connections. They have been a demonstrably valuable tool for modelling a variety of natural and social systems ([6], [7], [8]), and their applicability to machine learning (within the context of hidden Markov models) was already demonstrated by some of the authors [9], managing to achieve high performance on a sparse breast cancer classification dataset involving gene expression and methylation data.

The network design process is initiated by appropriately partitioning the input data—this may be done either manually (by exploiting existing domain knowledge) or through an unsupervised pre-training step, which will determine which (not necessarily disjoint) fragments of the input data are more likely to constructively influence one another. Afterwards, a cross-modal CNN is constructed such that a separate CNN *superlayer* is dedicated to each partition of the input data, attempting to learn the target function from its partition only. The purpose of the partitioning is to help the constituent CNNs become powerful predictors while requiring a smaller dimensionality of the input data, by allowing them access to
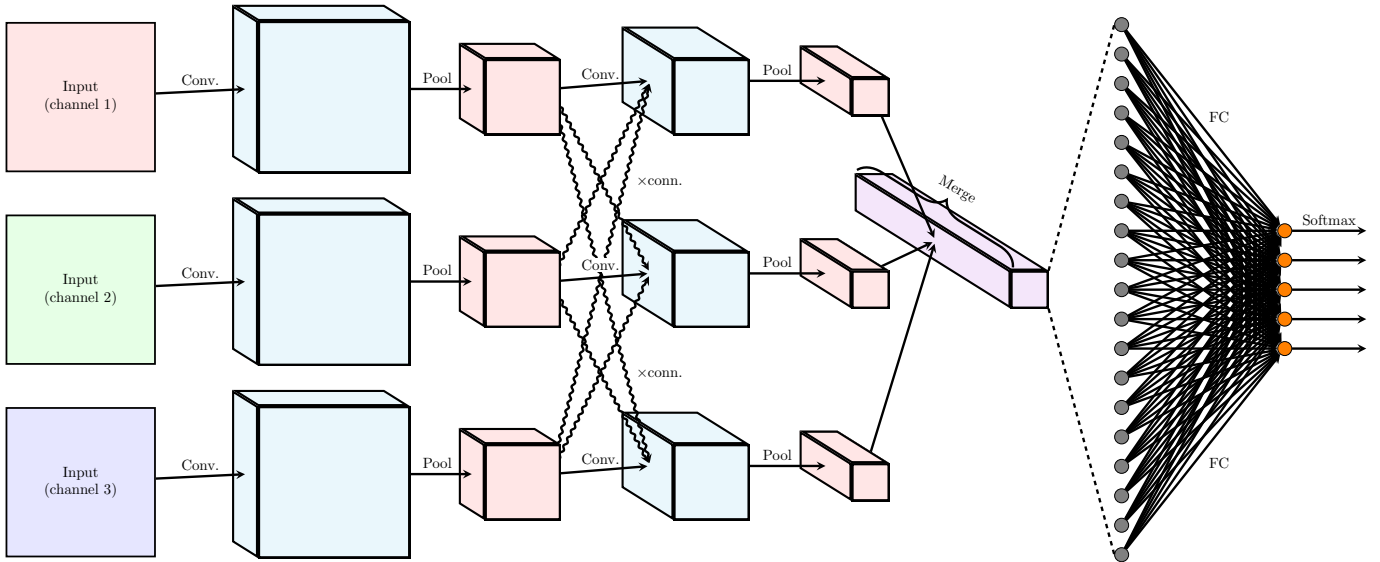
Fig. 1. Diagram of a simple cross-modal CNN for image classification, generated from a baseline CNN of the form $[Conv \to Pool] \times 2 \to FC \to Softmax$. Each of the three channels (RGB/YUV) of the input image receives its own CNN superlayer, with cross-connections inserted after the pooling operation, and full weight sharing in the fully connected layers. A more in-depth view of a potential cross-connection layout is provided by Figure 2.

those parts of the input which are most significantly related to each other in the context of the predictions that need to be made.

Finally, the superlayers may be interconnected by any sort of (feedforward) cross-connection as is best seen fit, and they may be combined in arbitrary ways at the output stage to produce the final output. Similarly, at any stage the weights of the superlayers may be shared—the simplest case, which we will explore in our analysis, constitutes complete weight sharing of the fully connected layers at the tail of the networks. This construction is biologically inspired by *cross-modal systems* [10] within the visual and auditory systems of the human brain (which in turn inspired the development of CNNs)—wherein several cross-connections between various sensory networks have been discovered [11], [12].

To quantify the gains of this approach, our evaluation focusses on an already well-understood problem of coloured image classification, on established CIFAR-10/100 [13] benchmarks for which an abundance of data is available, so it is easier to investigate the effects of restricting the size of the training set on various CNN models. The partitioning of the input that we consider is per-channel—each of the three image channels will be an input to an individual superlayer, and these superlayers will have identical high-level architecture (differing only in the number of feature maps per hidden layer)—as illustrated by Fig. 1. This also allows for a simple approach to cross-connections; namely, after every downsampling (*pooling*) operation we allow for the feature maps to be exchanged between superlayers, after being passed through another convolutional layer (Fig. 2).

While this model in itself constitutes a committee of CNNs, it differs from most traditional ensemble applications in two key ways:

- An ensemble's constituent models typically exchange information only in the *output* stage, while the cross-modal framework allows for arbitrary (feedforward) information flow at any stage of the processing pipeline;
- Constituent models of an ensemble usually receive a full copy of the input each, while superlayers within a cross-modal neural network receive only a fraction of the input, allowing for a *decrease in degrees-of-freedom* of the model compared to an unrestricted network.

In fact, this can be taken a step further: one may consider *ensembles of cross-modal CNNs*, which may compound on benefits already given by X-CNNs themselves, on examples where the networks are potentially struggling to choose a proper class with sufficient confidence. As the X-CNN model can be observed as an ordinary CNN from a high level, *any* ensemble strategies that are found useful for CNNs should be useful for X-CNNs as well.

Lastly, it should be noted that our approach is not restricted to CNNs, but it is then easiest to scrutinise, as the trained parameters are bound to obey a certain spatial structure. In line with this, an entire section of this manuscript will be dedicated to analysing the learned convolutional kernels within an X-CNN, as well as visualising the inputs that would maximise the model's cross-connection activations.

III. MODEL ARCHITECTURES

For the purposes of evaluating our proposed architecture's performance, we have implemented two baseline CNN models—along with their cross-modal variants—in Keras [14] (with Theano [15] back-end). For purposes of reproducibility, in this
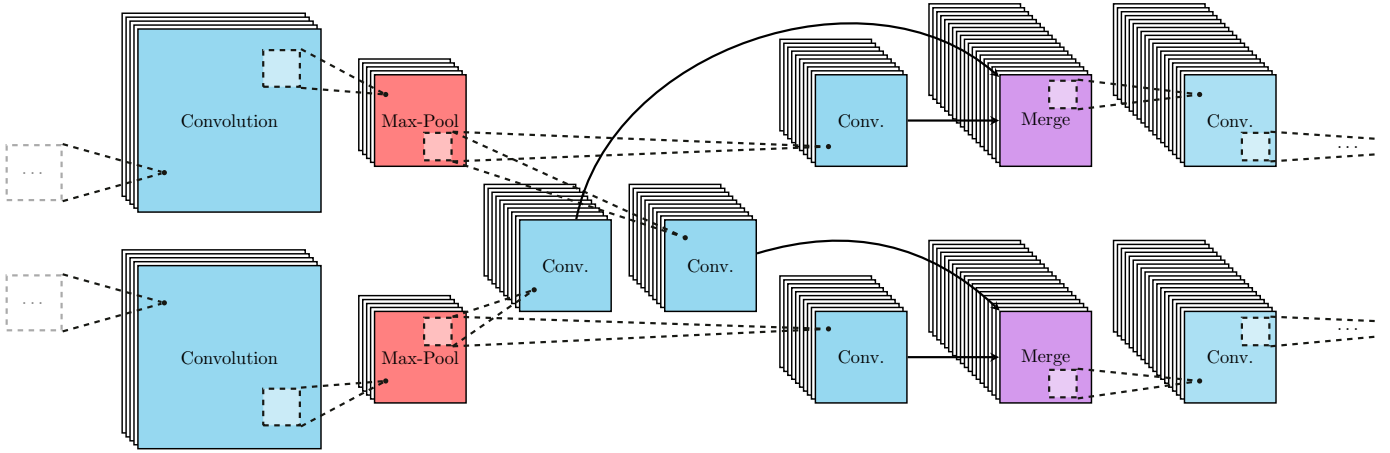
Fig. 2. Illustration of a single cross-connection segment within an X-CNN with two superlayers. After each pooling operation, we exchange the feature maps between the superlayers, after first passing them through an additional convolutional layer. We may also perform an additional intra-superlayer convolution before merging the feature maps in each superlayer via concatenation.

section we will expose their architectures and hyperparameters as used for the evaluation. The cross-modal variants' feature map counts have been altered in such a way as to make the overall number of parameters as close as possible to the baseline, making for fair evaluation with respect to degrees-of-freedom.

For both of the models used, we represent images in the YUV colour space. As a linear transformation from RGB, it should not have an impact on performance of the baselines, while it has the benefit of decoupling *luminance* from *chrominance*, allowing for a simpler analysis of cross-connections (and relating its learned kernels to human vision processes). We inject further domain knowledge into the model by favouring the CNN superlayer corresponding to the Y channel in terms of feature map counts (typically doubled compared to the U/V superlayers within the same hidden layer). This corresponds to the assumption that the majority of relevant information about an object is contained within its brightness channel, while colour usually represents auxiliary information.

A. KerasNet

Our initial model of choice represents a simple CNN with four convolutional ReLU [16] layers, followed by two fully connected layers, one of which is also ReLU. We will be referring to it as KerasNet throughout this manuscript as it is based on the Keras CIFAR-10 CNN example [17]. It represents a likely style of a "starting" model that one is going to attempt to apply on an image classification problem (without particular prior knowledge about it), perhaps especially bearing in mind that the training data may be sparse.

The architecture of the model, as well as its cross-modal variant (X-KerasNet) is outlined in Table I. Both models are trained for 200 epochs using the Adam SGD optimiser, with

TABLE I
ARCHITECTURES FOR KERASNET AND X-KERASNET

Output size	KerasNet $\sim 4.46\text{M param.}$	X-KerasNet $\sim 4.37\text{M param.}$
32×32	$[3 \times 3, 64] \times 2$	Y: $[3 \times 3, 32] \times 2$
		U/V: $[3 \times 3, 16] \times 2$
16×16	2 × 2 Max-Pool, stride 2	
		Y $\rightarrow$ Y: *identity*
		U $\rightarrow$ U: *identity*
		V $\rightarrow$ V: *identity*
		Y $\rightsquigarrow$ U/V: $[1 \times 1, 32]$
		U/V $\rightsquigarrow$ Y: $[1 \times 1, 16]$
	$[3 \times 3, 128] \times 2$	Y: $[3 \times 3, 64] \times 2$
		U/V: $[3 \times 3, 32] \times 2$
8×8	2 × 2 Max-Pool, stride 2	
1×1	Fully connected, 512-D	
	10/100-way softmax	

hyperparameters as described in [18], and a batch size of 32. Dropout [19] has been applied after both of the pooling operations (with $p = 0.25$) as well as after the first fully connected layer (with $p = 0.5$).

B. FitNet4

We decided to implement FitNet4 by Romero *et al.* [20] as our second baseline, representing a sophisticated CNN close to the state-of-the-art on CIFAR-10/100. We opted for this model as it is prominently featured in a variety of recent neural networks research ([21], [22]), and due to its design goal of being a "thin&deep" network, managing to keep its parameter count relatively low compared to many other successful models, and

Output size	FitNet4 ~ 2.75M param.	X-FitNet4 ~ 2.72M param.
32×32	$[3 \times 3, 32] \times 3$	Y: $[3 \times 3, 24] \times 3$
		U/V: $[3 \times 3, 12] \times 3$
	$[3 \times 3, 48] \times 2$	Y: $[3 \times 3, 36] \times 2$
		U/V: $[3 \times 3, 18] \times 2$
16×16	2×2 Max-Pool, stride 2	
		Y $\to$ Y: $[1 \times 1, 36]$
		U $\to$ U: $[1 \times 1, 18]$
		V $\to$ V: $[1 \times 1, 18]$
		Y $\rightsquigarrow$ U/V: $[1 \times 1, 12]$
		U/V $\rightsquigarrow$ Y: $[1 \times 1, 12]$
	$[3 \times 3, 80] \times 6$	Y: $[3 \times 3, 60] \times 6$
		U/V: $[3 \times 3, 30] \times 6$
8×8	2×2 Max-Pool, stride 2	
		Y $\to$ Y: $[1 \times 1, 60]$
		U $\to$ U: $[1 \times 1, 30]$
		V $\to$ V: $[1 \times 1, 30]$
		Y $\rightsquigarrow$ U/V: $[1 \times 1, 18]$
		U/V $\rightsquigarrow$ Y: $[1 \times 1, 18]$
	$[3 \times 3, 128] \times 6$	Y: $[3 \times 3, 96] \times 6$
		U/V: $[3 \times 3, 48] \times 6$
1×1	8×8 (global) Max-Pool	
	Fully connected, 500-D	
	10/100-way softmax	

therefore could still be a feasible first choice for handling a sparse dataset.

The FitNet4 consists of 17 convolutional 2-way maxout [23] layers, followed by two fully connected layers, the first of which is a 5-way maxout layer. The full architecture of this model—as well as its cross-modal variant (X-FitNet4)—is presented in Table II.

Both models are initialised using Xavier initialisation [24], and are then trained for 230 epochs using the Adam SGD optimiser with a batch size of 128. We have applied batch normalisation [25] to the output of each hidden layer to significantly accelerate the training procedure. L_2 regularisation with $\lambda = 0.0005$ has been applied to all weights in the model. Finally, dropout (with $p = 0.2$) was applied on the input, after every pooling operation, and after the fully connected maxout layer.

IV. EVALUATION

To verify our insights, we have utilised two well-known image classification benchmark datasets, CIFAR-10 and CIFAR-100 [13], for which an abundance of data is available (50000 training and 10000 testing examples). This makes it easier to study the behaviour of the considered CNNs as different fractions of the training data are discarded. We hypothesise that, at lower levels of data availability (up to a threshold),

our methodology will yield significant gains over an equivalent unrestricted CNN—and also that it will remain competitive at all higher training set sizes.

The validity of our claim is investigated by performing comparative evaluation, with the KerasNet and FitNet4 as baselines against X-KerasNet and X-FitNet4, respectively. In each individual test we evaluate the accuracy of these four models on the entire test set of 10000 samples, when the training routine is presented with only $p\%$ of the entire training dataset (chosen deterministically). The schedule for the tests is as follows:

- Initially test in increments of 5%, until reaching 20% (at which time the training and testing sets have equal sizes);
- Afterwards, test in increments of 10% until either reaching 50% or the accuracies of the two models get within 0.5% of each other (corresponding to a gain of ≤ 50 images properly classified), whichever is later;
- Specially, we always test on 1% (corresponding to a highly sparse environment with only 500 training images) and 100% of the training dataset.

The images are preprocessed by applying a single batch normalisation operation on them; we have found this to yield slightly better results compared to doing global contrast normalisation and ZCA whitening (the more common approach). Finally, given that it is, depending on the task, sometimes possible to significantly enhance results in a sparse environment by way of *data augmentation*, we have run all of the above tests *twice*—with and without random translations and horizontal reflections applied to the training images—providing insight as to whether data augmentation compounds the effects of a cross-modal architecture, and to what extent.

V. RESULTS AND DISCUSSION

The full evaluation results on the aforementioned tests are presented in Tables III–VI.

The results on tests without data augmentation are completely in line with the claim of Section IV; at sufficiently low training data sizes, both X-KerasNet and X-FitNet4 *significantly outperform* their respective baselines on the testing set, for both of the CIFAR-10/100 datasets.

For CIFAR-10, the threshold at which the baselines "catch up" (in terms of being able to manually learn the domain knowledge directly injected into their cross-modal variants) is at around $p = 40\%$, corresponding to 20000 training examples being available. Furthermore, on CIFAR-100, such a threshold is *never* reached, most likely due to the extreme sparsity of per-class examples making this problem particularly suitable for the X-CNN models; the only exception is the 1% scenario for FitNet4, where the data sparsity is probably too extreme (five examples/class) for such a deep model to reach its potential.

Regardless of when the threshold is surpassed, we report that the cross-modal CNNs will generally continue to have a slight

TABLE III
COMPARATIVE EVALUATION RESULTS ON CIFAR-10 WITHOUT DATA AUGMENTATION

Model \ p	1%	5%	10%	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
KerasNet	37.94%	53.82%	62.95%	67.39%	70.26%	74.39%	76.62%	78.55%	——	——	——	——	82.50%
X-KerasNet	**41.19%**	**57.84%**	**65.01%**	**68.25%**	**71.36%**	**74.79%**	**76.96%**	**78.57%**	——	——	——	——	**82.62%**
FitNet4	38.97%	56.78%	70.37%	75.07%	78.50%	81.95%	83.95%	85.22%	——	——	——	——	89.56%
X-FitNet4	**39.21%**	**60.57%**	**70.82%**	**76.09%**	**79.40%**	**83.36%**	**84.25%**	**86.14%**	——	——	——	——	**90.13%**

TABLE IV
COMPARATIVE EVALUATION RESULTS ON CIFAR-10 WITH DATA AUGMENTATION

Model \ p	1%	5%	10%	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
KerasNet	45.45%	67.01%	70.89%	78.83%	**80.97%**	**83.23%**	83.64%	85.02%	——	——	——	——	86.66%
X-KerasNet	**49.60%**	**69.28%**	**72.51%**	**78.96%**	80.58%	83.10%	**83.89%**	**85.37%**	——	——	——	——	**87.41%**
FitNet4	40.91%	**65.73%**	75.55%	80.85%	83.63%	86.23%	**88.30%**	89.11%	——	——	——	——	92.27%
X-FitNet4	**42.02%**	65.54%	**77.06%**	**81.33%**	**83.94%**	**86.41%**	88.13%	**89.37%**	——	——	——	——	**92.50%**

TABLE V
COMPARATIVE EVALUATION RESULTS ON CIFAR-100 WITHOUT DATA AUGMENTATION

Model \ p	1%	5%	10%	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
KerasNet	7.55%	15.10%	20.24%	24.76%	28.18%	32.43%	36.29%	38.61%	41.63%	44.10%	45.56%	46.26%	48.26%
X-KerasNet	**8.05%**	**16.45%**	**23.04%**	**26.91%**	**30.08%**	**35.39%**	**39.13%**	**41.88%**	**42.50%**	**45.96%**	**46.73%**	**48.25%**	**49.98%**
FitNet4	6.48%	16.84%	22.12%	28.30%	35.52%	39.28%	43.59%	49.69%	50.42%	55.83%	56.62%	58.00%	59.78%
X-FitNet4	**6.64%**	**18.73%**	**27.57%**	**33.59%**	**38.38%**	**45.53%**	**49.68%**	**52.21%**	**55.55%**	**57.22%**	**59.52%**	**60.87%**	**62.20%**

TABLE VI
COMPARATIVE EVALUATION RESULTS ON CIFAR-100 WITH DATA AUGMENTATION

Model \ p	1%	5%	10%	15%	20%	30%	40%	50%	60%	70%	80%	90%	100%
KerasNet	9.09%	24.68%	32.63%	38.64%	42.62%	47.64%	49.91%	52.46%	53.77%	54.26%	55.12%	55.42%	55.45%
X-KerasNet	**10.16%**	**27.15%**	**35.58%**	**42.05%**	**43.77%**	**48.80%**	**50.48%**	**54.25%**	**54.90%**	**55.33%**	**55.68%**	**56.82%**	**57.18%**
FitNet4	7.25%	17.94%	23.55%	29.24%	38.76%	48.07%	50.06%	56.01%	58.55%	59.80%	62.38%	63.60%	65.59%
X-FitNet4	**7.35%**	**20.39%**	**28.69%**	**37.86%**	**43.75%**	**50.48%**	**55.40%**	**57.92%**	**60.70%**	**62.76%**	**66.18%**	**66.27%**	**67.19%**

edge over the baselines—outperforming them on *all* of the full training dataset experiments, sometimes significantly. This naturally invites the conclusion that converting a CNN into a X-CNN (if allowed by the task) is always a reasonable step; it can yield significant benefits (the significance depending on the relation between the sparsity of the training dataset and the complexity of the baseline model), while rarely making performance significantly worse.

To further verify this claim, we have performed experiments on the full datasets (with and without augmentation) where we monitored how the testing accuracy evolves as a function of training epoch. The resulting plots are summarised in Figure 3; it is clear that the X-CNNs are at least as powerful as their baselines, even when the full training sets are available. Furthermore, it is possible to detect a narrow edge for the cross-modal models in the CIFAR-10 experiments, and a sig-

nificant edge in the CIFAR-100 experiments. The concluding remark is that even when the dataset under investigation is not very sparse, attempting to utilise a cross-modal variant of the considered models (if applicable) is a reasonable action, as it might yield noticeable returns in predictive power.

The analysis of the interplay between data augmentation and cross-modal networks on CIFAR-100 remains straightforward—the X-CNN models remaining consistently and significantly ahead of their baselines throughout the entire spectrum of training set sizes. On CIFAR-10, however, this is slightly more complicated; while the catch-up threshold gets expectedly decreased (to around $p = 20\%$), the behaviour of X-CNNs for smaller training set sizes does not always significantly compound the benefits of data augmentation. Specifically, at 5% of the training set the FitNet4 model manages to outperform X-FitNet4 (the roles do get reversed
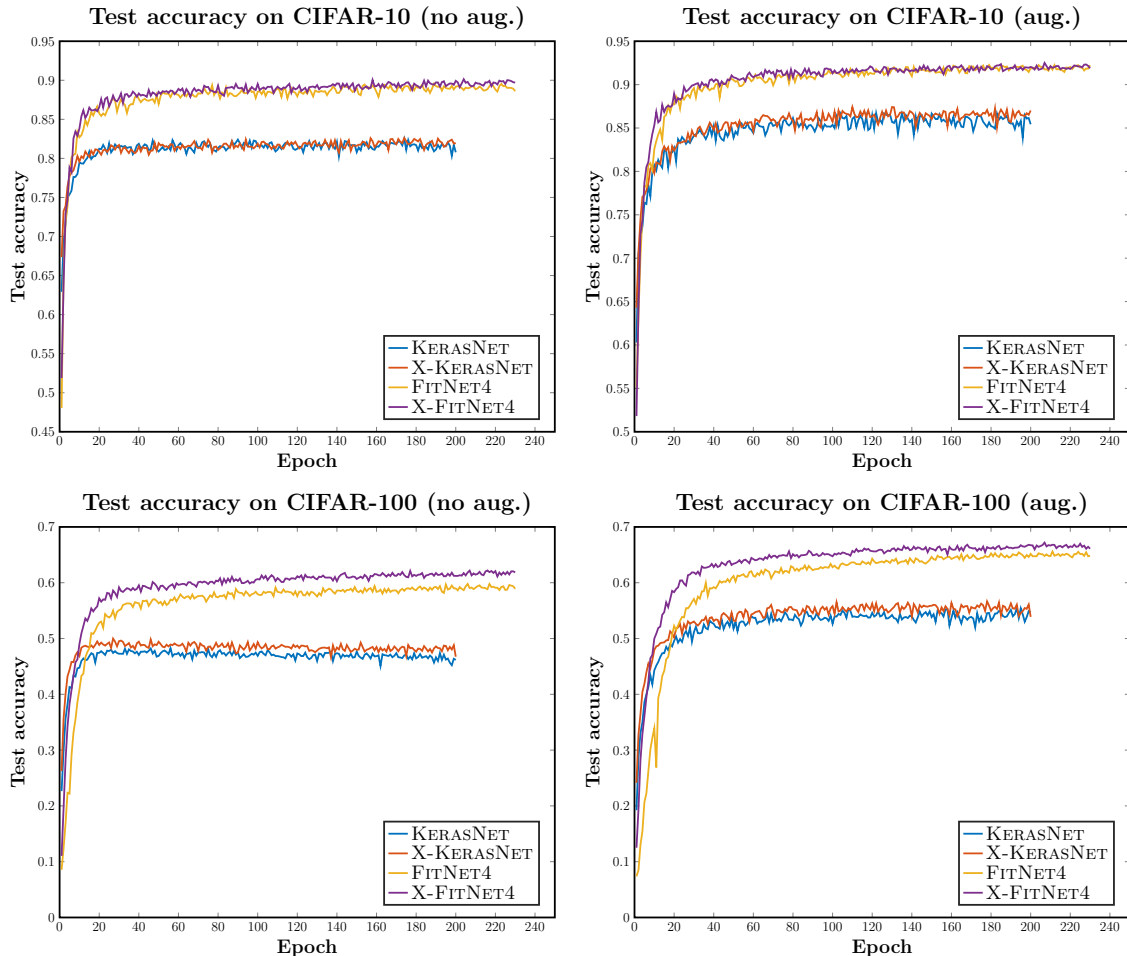
Fig. 3. Plots of the test accuracy of the four CNN models under consideration as a function of the number of training epochs, under 100% of the training set available. The experiments have been carried out on both CIFAR-10 and CIFAR-100, with and without data augmentation. The cross-modal CNNs are consistently competitive with their respective baselines across all four datasets, with a significant edge present for CIFAR-100.

starting from 10%, however). As a possible cause of this phenomenon, we note that, at this data availability level, both of the FitNet4 models are significantly inferior in performance to the KerasNet models, for which there is a significant benefit to the usage of X-CNNs. The takeaway lesson here is that, while the cross-modal architecture need not always compound nicely with data augmentation, an occurrence of such an event could signify that the baseline was not particularly suitable for properly accommodating data augmentation at this training set size in the first place. If this happens, one should attempt to use a more suitable/shallower CNN—the X-CNN variant should then produce the desired benefits.

Finally, we have taken advantage of some of the smaller training set sizes to perform *statistical significance tests*, typically scarce in deep learning literature. For training set sizes up to 15%, we trained the models five times (from different initial conditions) and then performed t-tests, choosing $p < 0.05$ as our significance threshold. Our findings show that, under these assumptions, the best-performing X-CNN

model's performance advantages are *statistically significant in all scenarios*, aside from the data-augmented CIFAR-10.

VI. CROSS-CONNECTION ANALYSIS

A key element of the X-CNN architecture are the cross-connection layers, as they enable information flow between individual channels. It will therefore be of interest to understand and visualise what is the mode of operation for these layers. All of the visualisations in this section correspond to the learned weights after fully training on 100% of CIFAR-10 with data augmentation.

We will first demonstrate that cross-connections inserted in the considered models, though being 1×1 convolutions, learn more complex functions than simple feature map passing. First, we note that the weights of a 1×1 convolutional layer may be represented as a 2D table that maps input channels to output channels (akin to an adjacency matrix, where columns are the input channels and rows are the output channels).
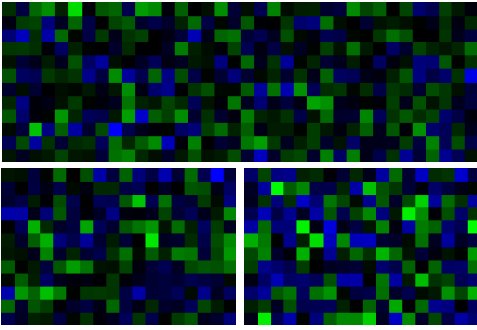
Fig. 4. Weight visualisation of the first-level cross-connection layer for the X-FitNet4 CNN. The columns correspond to input channels, while rows correspond to output channels. Green colour indicates a positive-weight connection between an input channel and an output channel, while blue colour indicates a negative-weight connection. The colour intensities are proportional to the absolute weight values. Top: Y ⤳ U/V (36 input channels, 12 output channels). Bottom, left-to-right: U ⤳ Y and V ⤳ Y (18 input channels, 12 output channels).
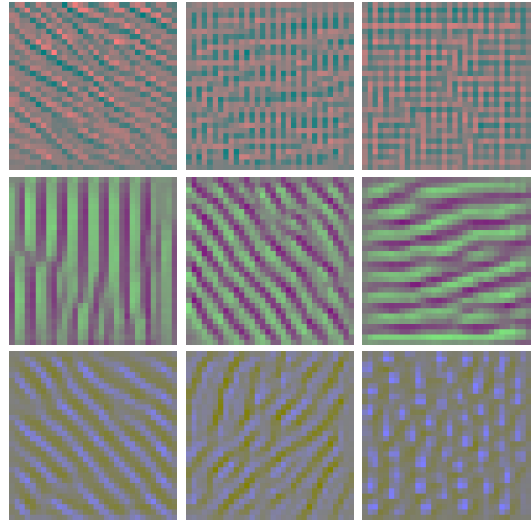

Fig. 5. Artificially generated images (from white noise) that cause strong activations of specific channels in the first cross-connection layer of the X-FitNet4 model. Top: Three channels from the Y ⤳ U/V cross-connections. Middle: Three channels from the U ⤳ Y cross-connections. Bottom: Three channels from the V ⤳ Y cross-connections.

Rather than displaying the raw table values, we decided to visualise weights in a *heatmap* style; Figure 4 showcases this visualisation for the first cross-connection layer of X-FitNet4. Green colours indicate that an input channel has a positive connection weight to the respective output channel while blue colours indicate negative weights. The colour intensities are proportional to the absolute weight values.

It can be seen that each output channel of the cross-connection layer is obtained through a *nontrivial weighted combination* of input channels. We hypothesise that the cross-connection layers selectively filter and combine input features that are more utilisable in another processing stream.

To delve deeper into what kinds of features the cross-connection layers are filtering, combining and passing, we applied layer-wise feature-map activation techniques proposed by Simonyan *et al.* [26]. This technique performs gradient ascent on a white-noise input image to maximise activations of a specific channel of feature maps at any of the layers within a pre-trained model. The objective function for gradient ascent is defined as

$$\mathbf{I}' = \underset{\mathbf{I}}{\arg\max} \, \Sigma(\mathbf{I}) - \lambda \|\mathbf{I}\|^2 \qquad (1)$$

where $\mathbf{I}$ is input image, $\Sigma(\mathbf{I})$ is the activation of the considered neuron when provided with $\mathbf{I}$ as input, and λ is a regularisation factor. After iterating for a number of gradient ascent steps, the original white-noise image will be modified into patterns that approximate the detection function of a specific neuron.

Lower-level convolutional layers are well-known to learn filters approximating Gabor wavelet filters that act as edge detectors, corner detectors, etc; we can confirm that in our experiments this has indeed been the case. For the first cross-connection layer of X-FitNet4, we have visualised a selection of channel activations in Figure 5. This visualisation indicates that the cross-connection layer is indeed passing combined lower-level features, such as the addition of horizontal and vertical stripes in the upper right image in the figure. We observe further that the pattern frequency for the Y channel's crossconnection layer is higher than the one for the U and V layers. This observation reflects the fact that the human vision system is able to detect higher frequency variations in intensity than chrominance. This is a solid indicator that the X-CNN architecture, when faced with an image classification task in the YUV colour scheme, is actually attempting to mimic human vision.

Our final analysis focusses on the X-KerasNet model, where we transformed feature maps of arbitrary depths into RGB images by a colour-mapping scheme. Figure 6 shows the feature maps of the inputs and outputs of the Y ⤳ U/V cross connections for representative images of the *truck* and *airplane* classes. These were easier to comparatively analyse on the X-KerasNet, as its cross-connection layers do not alter the number of feature maps, and therefore the same colour-mapping scheme remained meaningful for both. We observe that cross-connection output maps have background and some features emphasised, while other features de-emphasised—which further indicates that the cross-connection layers are performing more complex inter-superlayer feature integration than simply passing feature maps between superlayers.

VII. CONCLUSION

We have introduced cross-modal convolutional neural networks (X-CNNs), a novel architecture that decouples convolutional processing of (typically image-based) input partitions, while allowing for periodical information flow be-

Fig. 6. Visualisation of input and output feature maps of the Y ⤳ U/V cross-connection layer of X-KerasNet. Left: input images (*truck/airplane*). Middle: input feature maps to the cross-connection layer. Right: output feature maps of the cross-connection layer.

tween the processing pipelines, in order to achieve performance improvements in sparse data environments. We have applied this methodology on the popular CIFAR-10/100 image classification datasets for two baseline models, managing to significantly outperform them in low-data environments, while remaining competitive in high-data environments—outperforming them on *all* of the full-dataset experiments.

Aside from reinforcing the claim that the X-CNN architecture can only be beneficial to a baseline model (depending on the levels of training data sparsity, potentially highly significantly), we have further verified that the introduced cross-connection layers perform rather complex functions (thus they are not limited to simple feature map passing) and are capable of mimicking human vision processes—confirming that the biological inspiration behind such a model is justified.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, "Multilayer networks," *Journal of complex networks*, vol. 2, no. 3, pp. 203–271, 2014.

[6] M. De Domenico, C. Granell, M. A. Porter, and A. Arenas, "The physics of multilayer networks," *arXiv preprint arXiv:1604.02021*, 2016.

[7] E. Estrada and J. Gómez-Gardeñes, "Communicability reveals a transition to coordinated behavior in multiplex networks," *Physical Review E*, vol. 89, no. 4, p. 042819, 2014.

[8] C. Granell, S. Gómez, and A. Arenas, "Competing spreading processes on multiplex networks: awareness and epidemics," *Physical Review E*, vol. 90, no. 1, p. 012808, 2014.

[9] P. Veličković and P. Liò, "Molecular multiplex network inference using gaussian mixture hidden markov models," *Journal of Complex Networks*, 2015. [Online]. Available: http://comnet.oxfordjournals.org/content/early/2015/12/25/comnet.cnv029.abstract

[10] M. A. Eckert, N. V. Kamdar, C. E. Chang, C. F. Beckmann, M. D. Greicius, and V. Menon, "A cross-modal system linking primary auditory and visual cortices: Evidence from intrinsic fmri connectivity analysis," *Human brain mapping*, vol. 29, no. 7, pp. 848–857, 2008.

[11] A. L. Beer, T. Plank, and M. W. Greenlee, "Diffusion tensor imaging shows white matter tracts between human auditory and visual cortex," *Experimental Brain Research*, vol. 213, no. 2, pp. 299–308, 2011. [Online]. Available: http://dx.doi.org/10.1007/s00221-011-2715-y

[12] W. Yang, J. Yang, Y. Gao, X. Tang, Y. Ren, S. Takahashi, and J. Wu, "Effects of sound frequency on audiovisual integration: An event-related potential study," *PLoS ONE*, vol. 10, no. 9, pp. 1–15, 09 2015. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0138296

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[14] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

[15] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[17] F. Chollet, "Keras cnn example for cifar-10," github.com/fchollet/keras/blob/master/examples/cifar10_cnn.py, 2015.

[18] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.

[21] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

[22] D. Mishkin and J. Matas, "All you need is a good init," *arXiv preprint arXiv:1511.06422*, 2015.

[23] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 1319–1327.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.