Difficulty within Deep Learning Object-Recognition Due to Object Variance*

Qianhui Althea Liang[1]✉ and Tai Fong Wan[1]

SCSE, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798
{qhliang, twan002}@ntu.edu.sg

Abstract. It is one of the areas where deep learning models demonstrate possible performance bottleneck to learn objects with variations rapidly and precisely. We research on the variances of visual objects in terms of the difficulty levels of learning performed by deep learning models. Multiple dimensions and levels of variances are defined and formalized in the form of categories. We design "variance categories" and "quantitative difficulty levels" and translate variance categories into difficulty levels. We experiment how multiple learning models are affected by the categorization of variance separately and in combination (of several categories). Our experimental analysis on learning of the dataset demonstrates not only the expected way of utilization of variance of the data differs from models, the amount of learning or information gained for each data fed into models also varies significantly. Our results suggest it matters to search for a possible key representation of the "invariance" part of objects (or of the respective cognitive mechanism) and for the pertinent elements and capabilities in the deep learning architectures. It can be used to make the learning models a match to humans on complex object recognition tasks.

Keywords: Deep Learning · Variance · Difficulty in Deep Learning · Object Recognition.

1 Introduction

According to the studies from neuroscience and behavioural science, with regards to the existing implementation of Convolutional Neural Network (CNN), it lacks a certain mechanism to cater for an invariant representation [13] of a subject that has to be learnt. This, as a result, explains that CNN is unable to compete against the proficiency of human beings in tasks such as classifying or learning objects with greater variations. In the case of visual objects, it would seem that CNN has difficulties in learning the 3D variations, where the variations include the rotation of the 3D objects and changes to the surface of the objects.

Towards finding a remedy to overcoming such a difficulty, we have conducted research on the different variances of objects in terms of the difficulty levels of

* Research work is sponsored by Academic Research Funds of MOE Singapore. Authors thank the sponsorship.

learning performed by learning models. We study multiple dimensions and levels of variances and formalize them in the form of categories. Using a specially designed and carefully controlled data set, the impacts of the variance in the data set on learning in general deep learning are examined. We study how multiple learning models including DNN, CNN and RNN are affected by the categorization of variance separately and in combination (of several categories). The aim of our study is to come up with different levels of variations for the object recognition tasks or subjects, with respect to the variations of the neural network performance. Using several neural network stacks, our study will quantitatively evaluate the models with the measures that include complexity of neural network stack, duration of training time, accuracy and model output pattern. The study has been conducted in the domain of spatial image data. The novelty of our work lies on a definitive investigation on the difficulty levels of learning of various deep learning models in support of our notion of categories of variance in the objects and its impacts on learning. We claim that not only the expected way of utilization of variance of the data differs from models, the amount of learning or information gained for each data fed into models also varies significantly. Presenting limited amount of data but with maximized amount of information on variance, and/or of its categories, can help the learning models to gain progress notably. Our results suggest it matters to search for a possible key representation of the "invariance" part of objects (or of the respective cognitive mechanism) and for the pertinent elements and capabilities in the deep learning architectures in order to make them a match to humans on complex object recognition tasks.

2 Related Work

The major operations that have been studied in the image processing community and that result in the variances that we are interested in mostly include position, scale, pose variability [1]. Illumination, clutter and intra-class variability [2] are also considered. One study [3] had images that are generated from a variety of objects with an irrelevant background with randomized parameters following a defined range. The parameters controlled the position of the object (on the x and y axis), the rotation (on the x, y and z axis) and the size of object. With the amount or the degree of the operations of low, medium and high, the images were tested from volunteers. The higher the degree, the worse the performance of the volunteers. The tasks performed by the volunteers included object recognition (including facial recognition), object differentiation and facial differentiation, ordered from the best performed task. This shows to a strong extent that human proficiency at object recognition can be affected by the amount of variance introduced onto the object in the image.

Most of the related work focuses on designing datasets for tuning and understanding trained models' performance on the specific datasets and on their reliance to the amount of the image data. For example, in ObjectNet [4], the objects are placed in more varied backgrounds, and some objects have gone through operations of rotations and are taken from different viewpoints. They show that trained models do generally better on datasets than on real-world applications, to the extent of a 40-45% drop in performance. They demonstrate that models may possibly previously take advantage of bias in the available datasets and thus to perform well (e.g. objects are typically found with specific type of background [5]). As also shown [6], deep neural networks can easily fit the random labels. This may support the opinion that that trained models are very likely to show good results, misleading its capability on true generalization.

However, it also notable that most existing work has a strong reliance on huge datasets to reduce the correlation of features from the training set (to achieve generalization) [4], such that the invariant factors are learnt implicitly [1]. This might stem from when the well-known dataset ImageNet [8] was regularly used for its vast labelled dataset to train neural networks, where most models would be pre-trained and further fine-tuned to the task required. However, a large varied collection of data that may not relate to the actual task of the model, may not always bring the supposed benefits through pre-training.

When testing multiple models with different portions of the data [9], it is found that the reduction in classes or data provided to the models did not lead to a performance degradation in the CNN architecture. The study then suggests CNN could possibly be not as "data-hungry", which might imply that CNN could extract the needed generalization information for the given task with minimal data. Yet, with a more recent study into ImageNet pre-training [10], it shows that ImageNet pre-training may not be required as models trained from scratch do not necessarily per-form worse off. It is able to help models reach convergence faster but does not show performance improvement unless the target dataset to be trained on directly is small. This might then suggest that given a neural network, it can possibly perform averagely well given any form of data but would require task-relevant data to achieve better results.

Through the large amount of data provided in ImageNet, deep CNN was shown to be able to perform well solely through the use of supervised learning [11]. Deep CNN's performance degradation would be observed, had any of the convolutional layers were to be removed. As pointed out by them, "any given object can cast an infinite number of different images onto the retina" [12], thus the number of images required to represent this amount of variations present in the real world is far larger than the number of images our systems can store. Therefore, it is suggested [12] that synthetic images are better able to present this variation found in the transformation process. By having a greater control on the information present in our data, we can then rigorously test our models of different architecture to see how each internalises the information and thereby affecting the results.

3 Method

We have defined variances via variance categorization with its associated dimensions and degrees. We regulate the variances that are to be introduced in the sub-task of "figure-ground" segregation in a numerical range of numbers in order to further study the magnitude of variance impact. The smallest value zero (0) represents invariant or no variance, while the largest value five (5) represents maximal variance, each indicating a variance category. In section 3.1. we have included a brief introduction about such categorization. We then define the impacts of variance in terms of difficulty in learning. In particular, the difficulty in learning is quantified into difficulty levels, defined as "learning improvement" rate, i.e. $\mathrm{I_l}$. $\mathrm{I_l}$ is defined as the average improvement upon every epoch divided by the most recent loss value as the following. $\mathrm{I_l}$ uses the values from the dataset, and can be used to see how well the model explains the variance given such information.

$$I_l = \frac{Average(LossValueImprovementOfEveryEpoch)}{MostRecentEpoch'sLossValue} \tag{1}$$

We have derived the difficulty level treatment in deep learning corresponding to the variances presented to the neural network stacks in their learning process based on the following: There are obvious performance gaps among the varieties of variances [1] in learning. In addition, the gaps also demonstrate different magnitudes which may suggest a form of certain kind of hierarchy of performance differences. It is also observed that humans progressively perform worse at the task of object recognition when the variance was increased [1,2]. Therefore, our treatment of difficulty is to transfer it to different levels of categories.

On the other hand, among the three deep learning models (stacks) we have selected, CNN may experience greater difficulties in handling variant sub-tasks that is demonstrated by using very deep structure and longer time and bigger training set to complete the sub-tasks. We will use the output of neural network stacks or sub-structures to provide indications as to the magnitude of task difficulty from the perspective of the given neural network stack. The sub-structure can be as simple as a single layer of dense neurons or as complex as models with feedback mechanisms, where feedback is employed both in the training and executing phase. Recurrent Processing is such an example of the neural network stack with feedback mechanism that should perform reasonably well to handle variant sub-tasks. We will study what kind of indications are these, how these indications can be quantified (in ways such as loss pattern, consistency of accuracy pattern). With respect to the neural network architecture, the structure of the CNN model uses a bottom-up or feed-forward neural network stacks that do not form cycles. Conversely, the RNN model would be viewed as recurrent neural network stack that involves formation of cycles. Such that the complexities as viewed by one architecture such as CNN can be regarded as additional capabilities from the opposite perspective such as the RNN architecture where it processes the task from a different perspective. There could therefore exist multiple valid interpretation in the decomposition of the same task, so the goal is to then optimize and minimize the difficulty levels. Therefore, our treatment on difficulty also includes to find the relationship between the levels of categories and the learning capabilities of individual learning models on the sub tasks.

3.1 Datasets of Variances for Various NN Architectures

Here we present a brief summary of the development of our dataset first. The dataset is composed of 3430 images, each made up of a number of different objects, that are rendered with different backgrounds. They have been generated with the help of computer graphics software, in our case Pov-Ray [14]. The image creation utilizes ray-tracing to render 3D objects and place them within a 3D space. Therefore, the creation is done strictly in a XYZ coordinate system as well as light sources in the same system. Our dataset of pictures is designed to not only be "real-life" like, but also allow us to focus on the task of studying the variance of objects and its impacts on learning difficulty.

Given the variation level as defined previously, we are able to break down a given task into its components and give them a variant rating with the variant scale. Given a task of known variant components, we will then evaluate how the CNN and RNN stacks performs by observing the several indicators, where we define a numerical range of numbers (1 – 5) for the magnitude of difficulty. The indicators we will measure from the neural network model includes the training time of the model, accuracy of the model, output pattern of the model and complexity of the model (depth of model and number of units).

There are several operations that we have used to manipulate the variances of the objects during the creation of our data set. We will first try to map these operations to a corresponding variance category by following a theme of degrees of the visible change of the target object. These five categories and their mapped image processing operations are listed below and they will be analyzed individually: Category 5, the maximum variance category (Background Clutter), Category 4 (Surface Changing), Category 3 (Rotation), Category 2 (Scale), Category 1, minimum variance category (Movement). Based on the above categories, we study how difficult the different architectures of neural network are able to adapt and generalize the learning of specific tasks of a given domain. With this difficulty scale, we study how well the neural network models handle variant components of given tasks. For example, we may expect that both the CNN and RNN model will receive a difficulty rating of one (1) if tasked to differentiate geometric objects apart (such as a triangle from a circle). Where given the task of "figure-ground" segregation may be rated as a difficulty of five (5) for the CNN and a difficulty of three (3) for the RNN.

Given each variance category, we have also extended our dataset with another test dataset in order to test the impact of multiple variance categories. The focus of the work described here is to see if the effects of concurrent variance categories, whether they might be of additive or multiplicative in nature or more. Using Category 1 (Movement) as an example, the object translated without regards to the camera's position will result in movement with minor rotational effect.

To test the datasets, we will be focusing on the two types of neural network (NN) architectures. Top-down RNN model and bottom-up CNN model, while using a basic Dense (DNN) model as control.

The models will be tested with low complexity to prevent immediate overfitting, as the dataset is considerably small. We hope that the simpler the model, the more likely it is to capture interesting behavior during the testing phase.

For each NN architecture, we will record the loss value based on binary cross-entropy for both training and validation over 3 epochs. We will avoid using accuracy as a measure of the model because the dataset only contains positive samples (all pictures contain the target object) and models are likely able to blindly guess all positive and get a good result as a result eventually. The implementation for the models will be considerably different as well, and so using raw loss values will not be a good enough measure across the models.

3.2 Distribution of Data with Sensitivity Analysis

We have designed three different distributions with different amounts of the data in our data sets that are to be fed into the neural networks. We want to examine if there is any notable impact of feeding models with more data that are progressive more modified in terms of variance operations.

Model B Deep NN architectures trained on a single non-modified image. (The original image of the object).

Model B+[2-6] The dataset is split 5 parts, where for Model B+[2] NN architectures will receive the first 2 parts of the dataset and so on. For Model B+[6], Deep NN architectures trained will receive almost all of the data; a very small portion will be left out for testing.

Model B* Deep NN architectures are trained on a small amount of, heavily modified images. (Such as shifted to the corner of the picture in Movement, or rotated away from the camera in Rotation).

This shows how much information is required to achieve the desired performance and which architecture is able to perform better with the least amount of information provided. We theorize that if given a dataset of an insignificant variance category, the confidence of the model on the testing set should be explainable. For example, in Category 2 (Scaling), we would expect to see that the confidence of the model would be affected by some fixed explainable (linear or exponential) factor with respect to how to chair is scaled. Where in a dataset of a significant variance category, the confidence of the model on the increasing variance magnitude data will be difficult to explain.

4 Experiments

In our experiments, we have chosen to use CNN and RNN as the neural network architectures to be studied in our setting. The Dense (DNN) architecture is also studied and used as a comparison. We have performed extensively experiments

Table 1: Experimental verifications

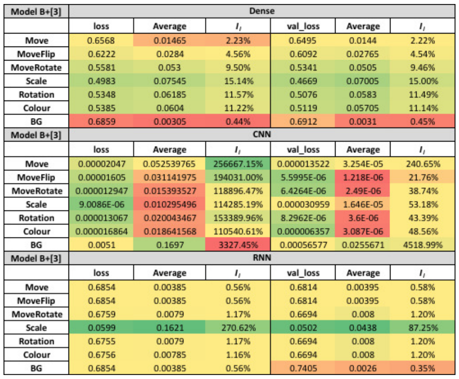
Variance Category	Data Distribution Model	Architecture	Confidence Graph
Move	B, B+[2-6], B*	CNN, RNN, DNN	✓
Scale	B, B+[2-6], B*	CNN, RNN, DNN	✓
Rotation	B, B+[2-6], B*	CNN, RNN, DNN	✓
Colour	B, B+[2-6], B*	CNN, RNN, DNN	✓
Background (BG)	B, B+[2-6], B*	CNN, RNN, DNN	✓

to study the difficulty of learning in terms of their $\mathrm{I_l}$ with the change of variance categories. Below we will only present some of our experiment results and use such results to support our findings.

In our previous study, we have proven that with the base data distribution model (Model B), our targeted network architectures are able to achieve an equivalently good training $\mathrm{I_l}$ over all variance categories. We, in this experiment, increase the amount of information on variance in the training set by presenting more pictures with possible variance, i.e. applying the distribution models from B+[2] all the way to B+[6] . We then examine how the training $\mathrm{I_l}$ changes as it trains on more such pictures. A selection of results of these experiments are shown in Figures 1, 2, 3, and 4.

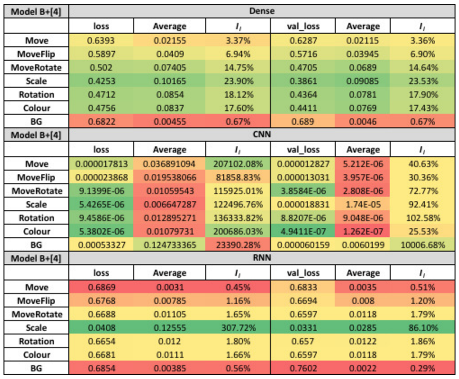
There is a trend for each targeted architecture that for a particular variance category it adapts in a way as seen in Figure 1. For example, in the Dense architecture, Rotation learns at the rate of 11.57% in Model B+[3] and increases to 25.11% in Model B+[6], as more information of variance are presented. Background that learns at with $\mathrm{I_l}$ of 0.44% in Model B+[3] and increases to 1.14% in Model B+[5] in Figure 3. We see that CNN architecture, when handling Background variance, learns with $\mathrm{I_l}$ of 3327.45% in Model B+[3] and increases to 244057.20% in Model B+[6]. While some of the validation $\mathrm{I_l}$ is 0% in CNN, we assume that the model may have perfectly fitted (or overfitted) the data, as we still see 38106.51% validation learning in Model B+[5] in Figure 3.

From our results, we can see that all the neural network models will fully utilize the information given them as expected. This can be observed by the notable difference in loss value from a model that did not receive it. For example, the loss rate of Rotation decreases from 0.0599 of Model B+[3] in Figure 1 to 0.0231 of Model B+[6] in Figure 4. From the results in Figure 1 to Figure 4, we see evidences that further support our claim that certain architectures are more adept at handling certain kinds of variance than others. This is particularly important when considering for commercial use, as we tend to push large amounts of data to models and assume that the invariant factors can be learnt implicitly. The different type of architectures might have different learning curves based on the type of information it receives, where we would hope to use the architecture that can exhibit an exponential learning curve. In Figure 5, models are trained on a small amount of selected images that have received the heaviest modification. The training learning values for Dense has increased all across, from the largest increase to 11.99% in Background and lowest increase to



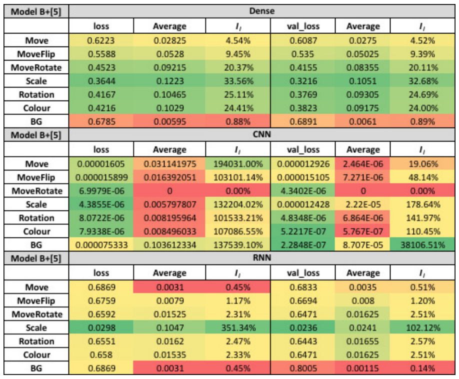
Model B+[3]	Dense					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6568	0.01465	2.23%	0.6495	0.0144	2.22%
MoveFlip	0.6222	0.0284	4.56%	0.6092	0.02765	4.54%
MoveRotate	0.5581	0.053	9.50%	0.5341	0.0505	9.46%
Scale	0.4983	0.07545	15.14%	0.4669	0.07005	15.00%
Rotation	0.5348	0.06185	11.57%	0.5076	0.0583	11.49%
Colour	0.5385	0.0604	11.22%	0.5119	0.05705	11.14%
BG	0.6859	0.00305	0.44%	0.6912	0.0031	0.45%
Model B+[3]	CNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.00002047	0.052539765	256667.15%	0.000013522	3.254E-05	240.65%
MoveFlip	0.00001605	0.031141975	194031.00%	5.5995E-06	1.218E-06	21.76%
MoveRotate	0.000012947	0.015393527	118896.47%	6.4264E-06	2.49E-06	38.74%
Scale	9.0086E-06	0.010295496	114285.19%	0.000030959	1.646E-05	53.18%
Rotation	0.000013067	0.020043467	153389.96%	8.2962E-06	3.6E-06	43.39%
Colour	0.000016864	0.018641568	110540.61%	0.000006357	3.087E-06	48.56%
BG	0.0051	0.1697	3327.45%	0.00056577	0.0255671	4518.99%
Model B+[3]	RNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
MoveFlip	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
MoveRotate	0.6759	0.0079	1.17%	0.6694	0.008	1.20%
Scale	0.0599	0.1621	270.62%	0.0502	0.0438	87.25%
Rotation	0.6755	0.0079	1.17%	0.6694	0.008	1.20%
Colour	0.6756	0.00785	1.16%	0.6694	0.008	1.20%
BG	0.6854	0.00385	0.56%	0.7405	0.0026	0.35%

Fig. 1: Difficulty results from model B+[3] by each architecture.



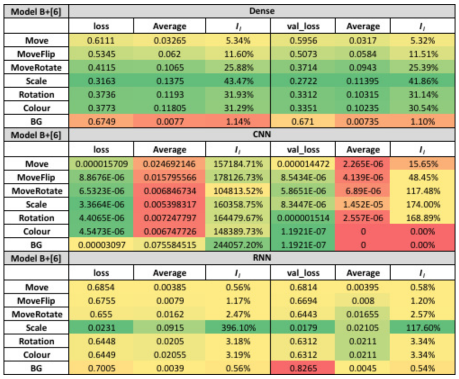
Model B+[4]	Dense					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6393	0.02155	3.37%	0.6287	0.02115	3.36%
MoveFlip	0.5897	0.0409	6.94%	0.5716	0.03945	6.90%
MoveRotate	0.502	0.07405	14.75%	0.4705	0.0689	14.64%
Scale	0.4253	0.10165	23.90%	0.3861	0.09085	23.53%
Rotation	0.4712	0.0854	18.12%	0.4364	0.0781	17.90%
Colour	0.4756	0.0837	17.60%	0.4411	0.0769	17.43%
BG	0.6822	0.00455	0.67%	0.689	0.0046	0.67%
Model B+[4]	CNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.000017813	0.036891094	207102.08%	0.000012827	5.212E-06	40.63%
MoveFlip	0.000023868	0.019538066	81858.83%	0.000013031	3.957E-06	30.36%
MoveRotate	9.1399E-06	0.01059543	115925.01%	3.8584E-06	2.808E-06	72.77%
Scale	5.4265E-06	0.006647287	122496.76%	0.000018831	1.74E-05	92.41%
Rotation	9.4586E-06	0.012895271	136333.82%	8.8207E-06	9.048E-06	102.58%
Colour	5.3802E-06	0.01079731	200686.03%	4.9411E-07	1.262E-07	25.53%
BG	0.00053327	0.124733365	23390.28%	0.000060159	0.0060199	10006.68%
Model B+[4]	RNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6869	0.0031	0.45%	0.6833	0.0035	0.51%
MoveFlip	0.6768	0.00785	1.16%	0.6694	0.008	1.20%
MoveRotate	0.6688	0.01105	1.65%	0.6597	0.0118	1.79%
Scale	0.0408	0.12555	307.72%	0.0331	0.0285	86.10%
Rotation	0.6654	0.012	1.80%	0.657	0.0122	1.86%
Colour	0.6681	0.0111	1.66%	0.6597	0.0118	1.79%
BG	0.6854	0.00385	0.56%	0.7602	0.0022	0.29%

Fig. 2: Difficulty results from model B+[4] by each architecture.



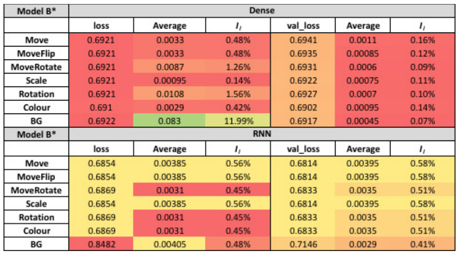
Model B+[5]	Dense					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6223	0.02825	4.54%	0.6087	0.0275	4.52%
MoveFlip	0.5588	0.0528	9.45%	0.535	0.05025	9.39%
MoveRotate	0.4523	0.09215	20.37%	0.4155	0.08355	20.11%
Scale	0.3644	0.1223	33.56%	0.3216	0.1051	32.68%
Rotation	0.4167	0.10465	25.11%	0.3769	0.09305	24.69%
Colour	0.4216	0.1029	24.41%	0.3823	0.09175	24.00%
BG	0.6785	0.00595	0.88%	0.6891	0.0061	0.89%
Model B+[5]	CNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.00001605	0.031141975	194031.00%	0.000012926	2.464E-06	19.06%
MoveFlip	0.000015899	0.016392051	103101.14%	0.000015105	7.271E-06	48.14%
MoveRotate	6.9979E-06	0	0.00%	4.3402E-06	0	0.00%
Scale	4.3855E-06	0.005797807	132204.02%	0.000012428	2.22E-05	178.64%
Rotation	8.0722E-06	0.008195964	101533.21%	4.8348E-06	6.864E-06	141.97%
Colour	7.9338E-06	0.008496033	107086.55%	5.2217E-07	5.767E-07	110.45%
BG	0.000075333	0.103612334	137539.10%	2.2848E-07	8.707E-05	38106.51%
Model B+[5]	RNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6869	0.0031	0.45%	0.6833	0.0035	0.51%
MoveFlip	0.6759	0.0079	1.17%	0.6694	0.008	1.20%
MoveRotate	0.6592	0.01525	2.31%	0.6471	0.01625	2.51%
Scale	0.0298	0.1047	351.34%	0.0236	0.0241	102.12%
Rotation	0.6551	0.0162	2.47%	0.6443	0.01655	2.57%
Colour	0.658	0.01535	2.33%	0.6471	0.01625	2.51%
BG	0.6869	0.0031	0.45%	0.8005	0.00115	0.14%

Fig. 3: Difficulty results from model B+[5] by each architecture.



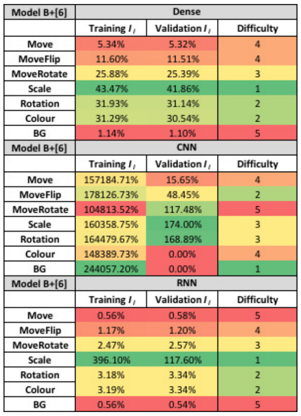
Model B+[6]	Dense					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6111	0.03265	5.34%	0.5956	0.0317	5.32%
MoveFlip	0.5345	0.062	11.60%	0.5073	0.0584	11.51%
MoveRotate	0.4115	0.1065	25.88%	0.3714	0.0943	25.39%
Scale	0.3163	0.1375	43.47%	0.2722	0.11395	41.86%
Rotation	0.3736	0.1193	31.93%	0.3312	0.10315	31.14%
Colour	0.3773	0.11805	31.29%	0.3351	0.10235	30.54%
BG	0.6749	0.0077	1.14%	0.671	0.00735	1.10%
Model B+[6]	CNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.000015709	0.024692146	157184.71%	0.000014472	2.265E-06	15.65%
MoveFlip	8.8676E-06	0.015795566	178126.73%	8.5434E-06	4.139E-06	48.45%
MoveRotate	6.5323E-06	0.006846734	104813.52%	5.8651E-06	6.89E-06	117.48%
Scale	3.3664E-06	0.005398317	160358.75%	8.3447E-06	1.452E-05	174.00%
Rotation	4.4065E-06	0.007247797	164479.67%	0.000001514	2.557E-06	168.89%
Colour	4.5473E-06	0.006747726	148389.73%	1.1921E-07	0	0.00%
BG	0.00003097	0.075584515	244057.20%	1.1921E-07	0	0.00%
Model B+[6]	RNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
MoveFlip	0.6755	0.0079	1.17%	0.6694	0.008	1.20%
MoveRotate	0.655	0.0162	2.47%	0.6443	0.01655	2.57%
Scale	0.0231	0.0915	396.10%	0.0179	0.02105	117.60%
Rotation	0.6448	0.0205	3.18%	0.6312	0.0211	3.34%
Colour	0.6449	0.02055	3.19%	0.6312	0.0211	3.34%
BG	0.7005	0.0039	0.56%	0.8265	0.0045	0.54%

Fig. 4: Difficulty results from model B+[6] by each architecture.



Model B*	Dense					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6921	0.0033	0.48%	0.6941	0.0011	0.16%
MoveFlip	0.6921	0.0033	0.48%	0.6935	0.00085	0.12%
MoveRotate	0.6921	0.0087	1.26%	0.6931	0.0006	0.09%
Scale	0.6921	0.00095	0.14%	0.6922	0.00075	0.11%
Rotation	0.6921	0.0108	1.56%	0.6927	0.0007	0.10%
Colour	0.691	0.0029	0.42%	0.6902	0.00095	0.14%
BG	0.6922	0.083	11.99%	0.6917	0.00045	0.07%
Model B*	RNN					
	loss	Average	I_l	val_loss	Average	I_l
Move	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
MoveFlip	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
MoveRotate	0.6869	0.0031	0.45%	0.6833	0.0035	0.51%
Scale	0.6854	0.00385	0.56%	0.6814	0.00395	0.58%
Rotation	0.6869	0.0031	0.45%	0.6833	0.0035	0.51%
Colour	0.6869	0.0031	0.45%	0.6833	0.0035	0.51%
BG	0.8482	0.00405	0.48%	0.7146	0.0029	0.41%

Fig. 5: Difficulty results from model B* by each architecture.



Model B+[6]	Dense		
	Training I_l	Validation I_l	Difficulty
Move	5.34%	5.32%	4
MoveFlip	11.60%	11.51%	4
MoveRotate	25.88%	25.39%	3
Scale	43.47%	41.86%	1
Rotation	31.93%	31.14%	2
Colour	31.29%	30.54%	2
BG	1.14%	1.10%	5
Model B+[6]	CNN		
	Training I_l	Validation I_l	Difficulty
Move	157184.71%	15.65%	4
MoveFlip	178126.73%	48.45%	2
MoveRotate	104813.52%	117.48%	5
Scale	160358.75%	174.00%	3
Rotation	164479.67%	168.89%	3
Colour	148389.73%	0.00%	4
BG	244057.20%	0.00%	1
Model B+[6]	RNN		
	Training I_l	Validation I_l	Difficulty
Move	0.56%	0.58%	5
MoveFlip	1.17%	1.20%	4
MoveRotate	2.47%	2.57%	3
Scale	396.10%	117.60%	1
Rotation	3.18%	3.34%	2
Colour	3.19%	3.34%	2
BG	0.56%	0.54%	5

Fig. 6: Learning improvement rates and difficulty from Model B+[6] ranked by the training learning rates.

0.14% in Scaling. Validation learning values have increased by very small values in comparison, where Background validation learning value did not improve at all. Whereas in CNN and RNN the training learning values have both increased by a larger comparison. For example, it is 0.48% in RNN. Such results from Model B* are encouraging as it shows that given very few pictures but with the information provided maximized, notable progress can be made. This is hopefully a notable point to raise as the existing trend in neural network training leans towards providing large chunks of data to facilitate implicit generalization learning. Our study might suggest that it might be possible to achieve a sufficient enough standard with specially tailored dataset, albeit not adequate standards given today's expectations of neural network.

Through our results in Figure 6, it might seem that inter-variance categories may not always necessarily be more difficult. That the Dense and RNN architecture would deal with the Movement with Rotation better than the Pure 2D movement dataset. However, it is also true that our results show the opposite for CNN. One possible reason might be that there might potentially by newer information when 2 operations are applied on a dataset, then whether the model is able to make use of the information to explain the data depends heavily on the architecture and how it views and encode the information. Our results might have been due to the dataset being notably small that the Dense and RNN are capable of memorizing certain aspects of the data instead of true learning. Unfortunately, our results also do not prove or disprove any relationship of inter-variance category to their parent categories. As we observe the values from the Dense and RNN Architecture as seen in Figure 6, the tasks can be grouped into different categories of similar impact. Scaling has the highest learning rate, followed by Rotational and Color in one group, the group of movement, and lastly background. Then comparing to the CNN architecture, it is clear that CNN performs significantly better at the background category, but the order of the other categories is debatable. Another notable point from Figure 5 is the inter-variance category Movement with Rotation, it would seem easier to the Dense and RNN architecture in terms of training learning value but not for the CNN architecture. CNN in this case then has better validation learning rate as it likely did not overfit the training data.

Among the outcome of our experiments, the confidence graphs of the CNN architectures with Model B are shown in Figures 7a-7e have the most reasonable pattern. Our results also shown that Dense architecture has some unexplainable fluctuations and RNN's being a bit flat for almost all graphs. Movement without Flip has the chair shifted from the center to the top left corner, yet the graphs have notable peaks and troughs that need further explanation. The Movement with Flip shows clearly in the center a clear drop, but the same pattern persists. Background has various local peaks and trough depending on the additional object accompanying the target object, the subsequent spike is due to later images where the target object is accompanied by a background scene.

The confidence graphs see higher variance impacts with some tasks. Background has various local peaks and trough depending on the additional object accompanying the target object, the subsequent spike is due to later images where the target object is accompanied by a background scene. Scaling is done by alternating between larger and smaller. Therefore, we can see a slight trend upwards and downwards while observing overlapping. Interestingly, the confidence goes up when the target object is scaled upwards and goes down when scaled downwards. This might suggest that scaling down objects causes some loss of information that the model uses. Movement with Rotation is steadily moving downwards. The rotating angle in the transition has helped; however, this graph can be studied further to be more explainable. Rotational has the target object rotated such that it returns to original position at the center and at the end. We can see that the confidence trends towards these two points and trending downwards when the rotation is further away from the original positions. Color increases somewhat in trend with brightness, the more visible the target object is, the more confident the model is. As the target object starts as black color (same as the background), and we see that when two of the RGB values are zero, the model drops back to the lowest point.

We have also presented the selected results of our study of the learning gain per data from all our distribution data models in Figure 8. To perform this study, we took all the learning values and divided by the number of images there were in each dataset. Throughout Model B to Model B+[6] for Dense, this might be the expected pattern most would have in mind if asked how Dense treats its information, all information given is equally important. Or possibly it just reflects the trajectory of the loss function towards a local minima. A similar story is seen in RNN except that it has a higher learning gain per data for the Scaling dataset. Figure 8 shows a spike in learning at Model B+[2] followed with a rather consistent decrease all the way till Model B+[6] with the exception of the Background dataset. The vital information was sufficient for CNN at Model B+[2] and it did not have much else to learn. Unlike the background dataset, especially in Model B+[5] and B+[6], where the images with background scene are added are likely the causes of the spike in values. Our study thus manages to suggest that there seems to be a mechanism in the architectures that models take advantage of to better analyses and explain the data in the domain of images.

To summarize, the described outcome of our experiments in this section will help in critical decision-making processes. Understanding the nature of variances and their impact on learning as variance categorization in general and in various deep learning network architectures results in explainable learning output, and will be highly desired. Further, we also expect that our results will be used as a guidance for researchers to design a mechanism to cater for better learning efforts. This shall be achieved by having learning models to acquire more intrinsic and higher responsiveness to the data but also their variance in their respective learning processes. Our experiment results have also suggested that it is likely such required mechanisms for every architecture may differ.

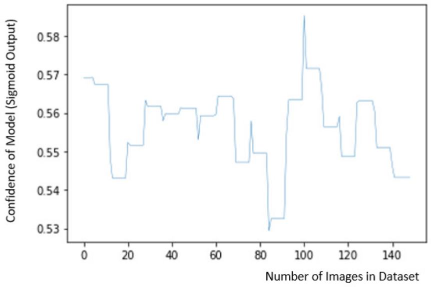


Fig. 7: Confidence for Movement.

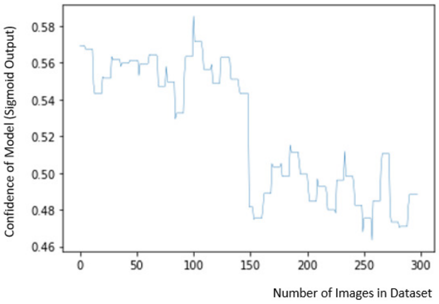


Fig. 8: Confidence for MoveFlip.

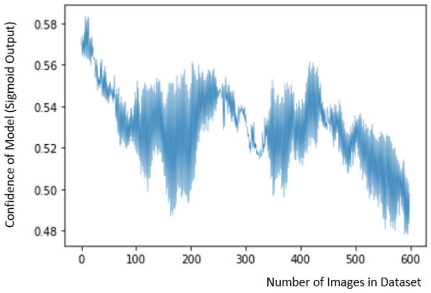


Fig. 9: Confidence for MoveRotation.

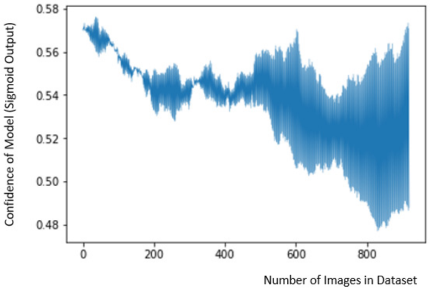


Fig. 10: Confidence for Scaling.

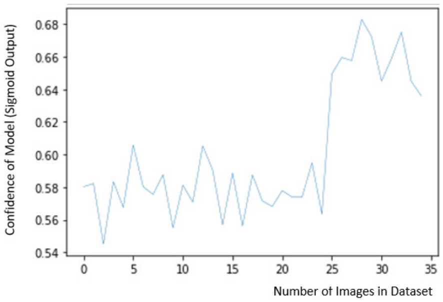
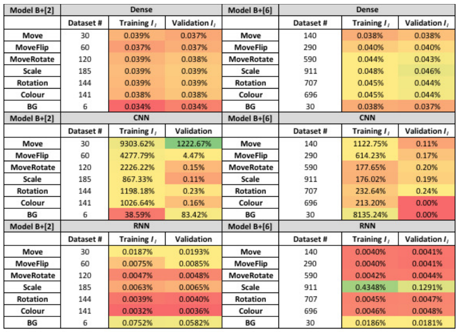


Fig. 11: Confidence for Background.



Model B+[2]	Dense			Model B+[6]	Dense		
	Dataset #	Training I_l	Validation I_l		Dataset #	Training I_l	Validation I_l
Move	30	0.039%	0.037%	Move	140	0.038%	0.038%
MoveFlip	60	0.037%	0.037%	MoveFlip	290	0.040%	0.040%
MoveRotate	120	0.039%	0.038%	MoveRotate	590	0.044%	0.043%
Scale	185	0.039%	0.039%	Scale	911	0.048%	0.046%
Rotation	144	0.039%	0.039%	Rotation	707	0.045%	0.044%
Colour	141	0.038%	0.038%	Colour	696	0.045%	0.044%
BG	6	0.034%	0.034%	BG	30	0.038%	0.037%
Model B+[2]	CNN			Model B+[6]	CNN		
	Dataset #	Training I_l	Validation		Dataset #	Training I_l	Validation I_l
Move	30	9303.62%	1222.67%	Move	140	1122.75%	0.11%
MoveFlip	60	4277.79%	4.47%	MoveFlip	290	614.23%	0.17%
MoveRotate	120	2226.22%	0.15%	MoveRotate	590	177.65%	0.20%
Scale	185	867.33%	0.11%	Scale	911	176.02%	0.19%
Rotation	144	1198.18%	0.23%	Rotation	707	232.64%	0.24%
Colour	141	1026.64%	0.16%	Colour	696	213.20%	0.00%
BG	6	38.59%	83.42%	BG	30	8135.24%	0.00%
Model B+[2]	RNN			Model B+[6]	RNN		
	Dataset #	Training I_l	Validation		Dataset #	Training I_l	Validation I_l
Move	30	0.0187%	0.0193%	Move	140	0.0040%	0.0041%
MoveFlip	60	0.0075%	0.0085%	MoveFlip	290	0.0040%	0.0041%
MoveRotate	120	0.0047%	0.0048%	MoveRotate	590	0.0042%	0.0044%
Scale	185	0.0063%	0.0065%	Scale	911	0.4348%	0.1291%
Rotation	144	0.0039%	0.0040%	Rotation	707	0.0045%	0.0047%
Colour	141	0.0032%	0.0036%	Colour	696	0.0046%	0.0048%
BG	6	0.0752%	0.0582%	BG	30	0.0186%	0.0181%

Fig. 12: Learning gain per data.

5 Conclusion

In this paper, we study how variance of datasets are translated into difficulties in deep learning. We experiment on how different deep learning models and their architectures respond to such variance in their respective learning activities in terms of the learning performance of object recognition. We claim that the expected way of utilization of variance information of datasets in our specific learning task and learning environment differs from models. The amount of learning or information gained for each data fed into models also varies significantly in our setting. The investigation of tasks other than object recognition is being conducted and we hope to extract common elements to shed light on a possible invariance representation that can help machine to match the performance of human beings in similar learning tasks.

References

1. Amit, Y., Felzenszwalb, P.: Object Detection, p. 537–542 (2014)
2. DiCarlo, J.J., Zoccolan, D., Rust, N.C.: How does the brain solve visual object recognition? Neuron, 73(3): p. 415–434 (2012)
3. Majaj, N.J., Hong, H., Solomon, E., DiCarlo, J.: Simple Learned Weighted Sums of Inferior Temporal Neuronal Firing Rates Accurately Predict Human Core Object Recognition Performance. The Journal of Neuroscience, 35(39): p. 13402 (2015)
4. Barbu, A., et al.: ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. in Neural Information Processing Systems. (2019)
5. Zhu, Z., Xie, L., Yuille, A.L.:Object recognition with and without objects. arXiv e-prints (2016)
6. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. International Conference on Learning Representations. OpenReview.net, Toulon, France (2017)
7. Geirhos, R., et al.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. International Conference on Learning Representations. OpenReview.net (2019)
8. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Li, F.: ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
9. Huh, M., Agrawal, P., Efros, A.: What makes ImageNet good for transfer learning? arXiv e-prints (2016)
10. He, K., Girshick, R., Dollar, P.: Rethinking ImageNet Pre-Training. 2019 IEEE International Conference on Computer Vision, pp. 4917-4926 (2019)
11. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet Classification with Deep Convolutional Neural Networks. Association for Computing Machinery. Communications of the ACM, 60(6), 84–90 (2017)
12. Pinto, N., et al. : Why is Real-World Visual Object Recognition Hard? (Real-World Visual Object Recognition). PLoS Computational Biology, 4(1), e27 (2008)
13. Kheradpisheh, S., et al.: Deep Networks Can Resemble Human Feed-forward Vision in Invariant Object Recognition. Sci Rep 6, 32672 (2016)
14. Pov-Ray Website, `http://www.povray.org/download/`.