From On-Road to Off: Transfer Learning Within a Deep Convolutional Neural Network for Segmentation and Classification of Off-Road Scenes

Christopher J. Holder[1,2(✉)], Toby P. Breckon[2], and Xiong Wei[1]

[1] Institute for Infocomm Research, Singapore, Singapore
[2] School of Engineering and Computer Sciences, Durham University, Durham, UK
c.j.holder@durham.ac.uk

Abstract. Real-time road-scene understanding is a challenging computer vision task with recent advances in convolutional neural networks (CNN) achieving results that notably surpass prior traditional feature driven approaches. Here, we take an existing CNN architecture, pre-trained for urban road-scene understanding, and retrain it towards the task of classifying off-road scenes, assessing the network performance within the training cycle. Within the paradigm of transfer learning we analyse the effects on CNN classification, by training and assessing varying levels of prior training on varying sub-sets of our off-road training data. For each of these configurations, we evaluate the network at multiple points during its training cycle, allowing us to analyse in depth exactly how the training process is affected by these variations. Finally, we compare this CNN to a more traditional approach using a feature-driven Support Vector Machine (SVM) classifier and demonstrate state-of-the-art results in this particularly challenging problem of off-road scene understanding.

1 Introduction

Scene understanding is a vital step in an autonomous vehicle processing pipeline, but this can be especially challenging in an off-road, unstructured environment. Knowledge about upcoming terrain and obstacles is necessary for deciding on the optimum path through such an environment, and can also be used to inform vehicle driving parameters to improve traction, efficiency and maximise passenger comfort and safety.

Whole scene understanding is a well-discussed problem with applications in many domains [1, 2]. Recent contributions have used convolutional neural network (CNN) based approaches to achieve state-of-the-art results [3], while approaches combining hand-crafted features with linear classifiers have been somewhat side-lined [4].

Work in the domain of scene understanding for autonomous vehicles has followed this trend [5, 6], however there is very little work applying deep-learning techniques to the more challenging off-road environment. This paper aims to assess the applicability to such an environment of a state-of-the-art CNN architecture that was originally designed and trained to perform per-pixel classification on urban road scene images [6].

G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part I, LNCS 9913, pp. 149–162, 2016.
DOI: 10.1007/978-3-319-46604-0_11

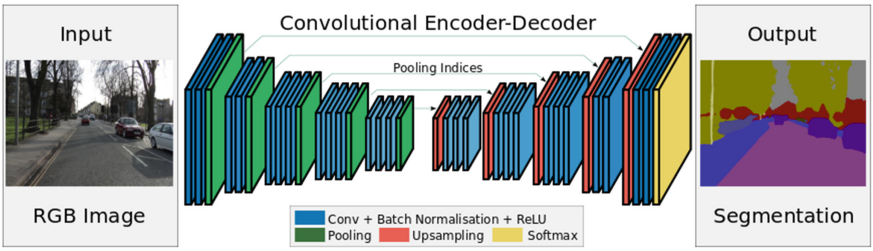

Fig. 1. Architecture of the Segnet convolutional neural network [6]. The encoder network, consisting of convolution and pooling layers, is followed by a mirror-image decoder network, consisting of convolution and up-sampling layers

Within this work we perform transfer learning, taking a CNN architecture that has already been originally trained to classify a large, often more generic data set and re-training it from this initialization to a more specific or alternative task (for which data is often more limited). In this case, a CNN trained for urban street scene classification is subsequently re-trained with a smaller, more specialised data set of off-road scenes. The idea is that the weights learned on the larger data set act to build a set of generic image filters that can be easily adapted for the task of classifying the more specialised imagery used later [7]. Transfer learning is generally thought to be beneficial when training with a small specialised data set or when the time to train a new network from scratch is not available, so we investigate the effects of data set size and training time on the classification performance of networks that have performed different amounts of pre-training or no pre-training at all.

Most existing work in the area of off-road classification does not make use of deep-learning techniques: the approach described in [8] aims to classify different parts of a colour image of an off-road scene using Gaussian Mixture Models, while the approach outlined in [9] uses a combination of features from colour imagery and 3D geometry from a laser rangefinder to classify the different parts of an off-road scene.

For comparison with our CNN based approach, we use a method based on the state-of-the-art object category retrieval work in [10]: dense gradient features are clustered to build a histogram encoding that is fed into a support vector machine (SVM) [11] for classification.

2 Methodology

We primarily propose a convolutional neural network approach and compare this to a secondary support vector machine approach for relative performance evaluation.

2.1 CNN Architecture

The convolutional neural network architecture we use is nearly identical to the ‘Segnet’ architecture described in [6], with only minor changes made to the final layer of the

network in order to output eight classes and to adjust the class weightings for our off-road data set. Similar network architectures exist [3], however we down-selected Segnet due to the focus of its creators on autonomous vehicle applications and its ability to perform real-time classification.

Fig. 2. An example image from the Camvid dataset along with its annotations

The Segnet architecture is visualised in Fig. 1. It is comprised of a symmetrical network of thirteen 'encoder' layers followed by thirteen 'decoder' layers. The encoder layers correspond to the convolution and pooling layers of the VGG16 [12] object classification network, while the decoder layers up-sample their input so that the final output from the network has the same dimensions as the input image. During the encoding phase, each pooling layer down-samples its input by a factor of two and stores the location of the maximum value from each 2×2 pooling window. During the decoding phase, these locations are used by the corresponding up-sampling layer to populate a sparse feature map, with the convolution layers on the decoder side trained to fill the gaps. This technique facilitates full pixel-wise classification to be achieved in real-time, making Segnet an ideal architecture for use in further autonomous vehicle applications.

2.2 CNN Training

We begin by training the network on the Camvid dataset [13] that was used by the original authors to assess Segnet. By training on a large, well labelled dataset that has already been shown to work well with this network architecture we can ensure that our network learns a set of weights that are relevant to a vehicular scene understanding task. We then perform transfer learning, retraining the network on our own off-road data so that it can adjust its weights to better suit an off-road environment and discriminate between the classes present in these scenes.

The benefits of transfer learning are in the ease with which an existing trained network can be adapted to a new specialised task. The time taken to train the network and learn optimum weights should be greatly reduced when compared to a network being trained from scratch with randomly initialised weights. In cases where the

specialised data set is small or only partially labelled, a network trained from a random starting point may never achieve satisfactory results, however by performing the bulk of training with a larger set of data and only utilising the specialised data set for the last few iterations, a better outcome can be achieved [7].

In our case, the initial training data consists of 367 labelled images of urban street scenes from the Camvid data set, resized to a resolution of 480 × 360. An example image from the dataset, along with its annotations, can be seen in Fig. 2. The original authors chose eleven pixel classes for the Segnet classification task, *{sky, building, pole, road marking, road, pavement, tree, sign, fence, car, pedestrian, and bicycle}*. As the network architecture performs classification of every pixel, this gives us up to 172,800 samples per image, or 63,417,600 samples in total. In practice, the total is slightly less than this as some images have pixels that do not fit into any of the eleven original classes and are labelled 'void'.

Fig. 3. An example image from our off-road dataset next to its partially labelled training image

Our off-road data consists of 332 images captured by a vehicle mounted camera driven at two different off-road driving facilities encompassing a variety of environments, which we split into roughly 90 % training data and 10 % test data, giving us 295 training images. For the CNN, our images have been resized to the same resolution of 480 × 360 as the Camvid images. We identified 8 pixel class labels for our off-road data set, *{sky, water, dirt, paved road, grass, foliage, tree, and man-made obstacle}*, 3 of which also existed in the Camvid data.

Fully labelling every pixel in even a small set of images can be very time consuming, so we only partially label our training images to assess whether good classification results can still be achieved without full labelling. Our labelling strategy consists of hand drawing a shape that is entirely contained by, but not touching the edges of, each image segment. Every pixel within that shape is then considered a member of the chosen class. Another reason this approach was chosen is the lack of clear boundaries to delineate classes in off-road scenes, for example when a muddy surface gradually gives way to gravel, or where long grass becomes foliage. This provided us with a total of 35,016,288 labelled pixels for training, with the rest of the pixels (roughly 31 % of the total) labelled as void so that the network would ignore

them. An example image from our dataset, along with its annotations, can be seen in Fig. 3. For testing our classification results, we use one set of 37 images labelled in the same manner, as well as another set of 4 fully labelled images. Figure 4 shows the partial and fully labelled versions of one of our test images.

To test the effects of transfer learning when the specialised training is carried out with a small data set, we train several versions of the network using different sized subsets of this data, one each containing 140, 70, 35, 17, and 8 of the original images.

Snapshots are taken at several points during the Camvid training, so that we can observe the effects of different amounts of pre-training. Seven versions of the network will be trained and assessed on our off-road dataset: one which has been randomly initialised with no prior training, along with networks trained for 1000, 2000, 5000, 10,000, 20,000 and 30,000 iterations on the Camvid data.

Our training is performed on an NVidia Tesla K40 GPU, taking roughly one hour per thousand training iterations.

Fig. 4. Partial and fully labelled versions of the same image from our off-road data set. Fully manually annotating an image like this can take a person several hours, while a partially annotated image can be created in a few minutes. In our partially labelled training set, 69 % of pixels are labelled

2.3 Support Vector Machine

For comparison, we will be training a SVM to classify the same data using dense gradient features, based on the approach used in [10] for object classification. For our approach we will be classifying image segments, as this will allow us to cluster feature points to build up a bag-of-words vocabulary. The segments in this case are those created manually while labelling the data, as in this case we are only interested in the performance of the classifier itself and so a perfect segmentation is assumed. In practice, an imperfect segmentation algorithm would be used, potentially leading to errors in the segmentation that could impact classifier performance.

To ensure enough local gradient information is available at each feature point, we use images with a resolution of 1280×720, higher than those used to train the CNN. The memory and time that would be required to train the CNN using images at this resolution would be infeasibly high, however by clustering our features before passing

them to the SVM, the size of data it uses to train and classify is constant per data sample regardless of image resolution.

Our labelled data set gives us 5664 labelled segments, which we split into 90 % for training data and 10 % for testing data. We ignore any samples too small to provide at least 50 feature points, leaving us with between 3000 and 4000 viable segments, depending on the feature grid density used.

We train a Support Vector Machine for a maximum of 20,000 iterations using a radial basis function to perform a grid search over the kernel parameter space.

Dense Feature Descriptors. A dense grid of feature points is computed for each segment. The grid density, g pixels between grid nodes in both x and y direction, is chosen empirically by testing values between 2 and 10 pixels. Generally, a denser grid should contain a greater amount of information at the expense of computation time, so a lower number should give better results in most cases.

The Speeded Up Robust Features (SURF) algorithm [14] is used to create a descriptor for each remaining grid node. A SURF descriptor computes Haar wavelet responses within a square region around the initial point, which are summed to produce a vector describing the intensity distribution of pixels within the region. This results in either a 128 or 64 dimension vector that describes the local texture. Empirically we found a 64 dimension vector to give better results at this task. Every SURF descriptor is computed at the same orientation of 0 radians with a radius of r pixels. r is chosen empirically after assessing classification results using a range of values from 2 to 20 pixels.

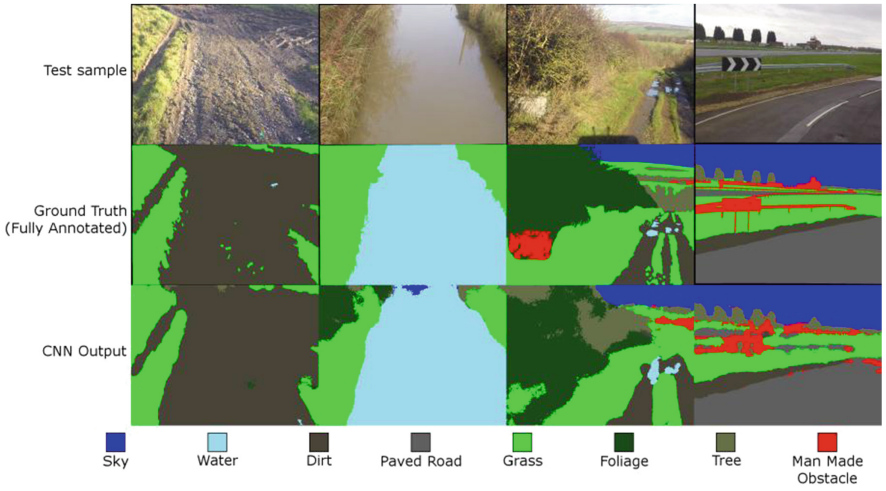

Fig. 5. Results from the CNN after 30,000 iterations of pre-training and 10,000 iterations of training with the full off-road dataset. The middle row shows the fully annotated test images for comparison, and class colour labels are shown below

Feature Encoding. Descriptors resulting from grid-wise feature extraction over a given segment are encoded into a fixed length vector for subsequent classification.

We use a histogram encoding (traditional bag-of-words [15]) approach: for histogram encoding, we first use K-means clustering to create a visual vocabulary, or bag of words, of K clusters within the 64 dimensional space of our SURF descriptors. For each segment, a histogram is computed accumulating the number of its SURF descriptors assigned to each cluster within the vocabulary. This histogram is normalised to provide a K-dimensional descriptor for the segment as the input feature vector to the SVM. The optimum value for K is chosen empirically, after testing values from 200 to 1600.

3 Results

We evaluate our classifiers using two sets of test data: a set of images partially labelled in the same manner as our training data, and a smaller set that are fully labelled (i.e. every single pixel in the image is labelled). The output layer of the CNN assigns a label to every pixel, while the SVM outputs a label for each segment. Figure 5 shows some example images with their respective CNN outputs and ground truth annotations.

Due to the lack of clearly defined boundaries in some areas of off-road scenes, there exist some pixels could have more than one correct label in terms of true ground truth. This should not have much effect on the partially labelled data, as boundary regions remain largely unlabelled, however this is likely to have a negative effect on classification results when testing against fully labelled data. To limit this effect, when deciding whether a pixel is correctly labelled we search for a match within a 5 pixel radius in the ground truth image. When testing with partially labelled data, a pixel is only labelled correctly if a match is found at its exact location in the ground truth image.

When discussing the CNN, unless stated otherwise, accuracy is defined as the number of correctly labelled pixels divided by the total number of labelled pixels in the

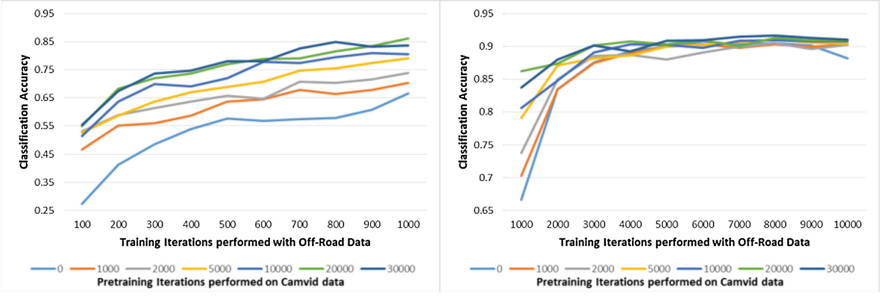

Fig. 6. Comparison of training progress for networks that have undergone different amounts of pre-training on the Camvid urban data set. We plot classification accuracy at every 100th iteration during training with our off-road data set for the first 1000 iterations, then at every further 1000th iteration until 10,000 iterations have been trained

test data. When discussing the SVM, accuracy is defined as the number of correctly labelled segments divided by the total number of labelled segments in the test data.

3.1 CNN with Partially Labelled Test Data

First we compare classification accuracy from training the network on our full off-road data set as well as smaller subsets thereof after different amounts of pre-training, and testing on our partially labelled test data set.

Table 1. Accuracy of the CNN on the Camvid test data at the points when snapshots are taken to perform transfer learning

Iterations trained	1000	2000	5000	10000	20000	30000
Classification accuracy	0.31	0.36	0.48	0.68	0.75	0.79

Pre-training Iterations. Table 1 shows the performance of the network on Camvid test data before any training with off-road data, with accuracy recorded at the six points from which transfer learning was to be performed. As the Camvid data is mostly fully labelled, we use the same measure of accuracy as we use with our fully labelled off-road test data set, wherein a label is deemed to be correct if it is within a 5 pixel radius of a similarly labelled pixel in the ground truth image.

These results demonstrate the network has rapid performance improvement over its first 10,000 training iterations, followed by a slower but consistent improvement in performance during later training iterations.

Figure 6 shows the results achieved by each pre-trained version of the network on our full data set. Each version of the network was trained for 10,000 iterations, with a snapshot taken and accuracy recorded first at every 100 iterations, then at every 1000 iterations.

The results show that the first few thousand iterations clearly benefit from transfer learning, with the networks that have performed a greater amount of pre-training generally performing better. However, by 5000 iterations of training, even the network

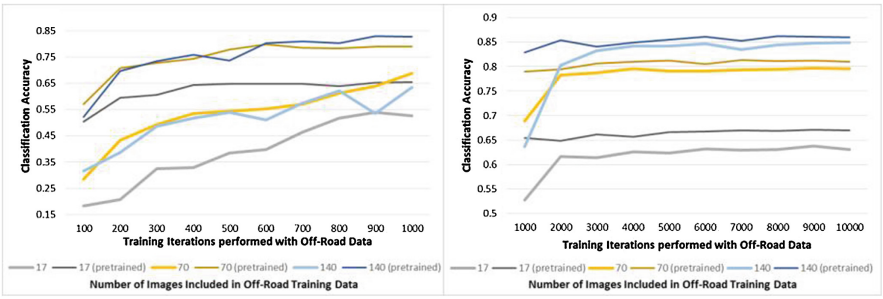

Fig. 7. Comparing pre-trained and non-pre-trained networks using different sized subsets of our off-road data set

initialised with random weights has achieved an accuracy of close to 0.9, beyond which there is very little improvement from any of the networks.

As the training continues, the networks pre-trained for longer give marginally better results. The highest accuracy achieved is 0.917, which comes after 8000 iterations of the network that was pre-trained for 30,000 iterations. The networks pre-trained for 20,000 and 30,000 iterations show very similar results throughout the training, suggesting a limit to the performance gains that can be achieved by pre-training.

Training was continued up to 20,000 iterations with each network, however this gave no further increase in accuracy and so only the first 10,000 iterations are shown.

It is interesting to note that our results surpass those achieved by their respective networks on the Camvid test data within a few hundred iterations, and then go on to perform significantly better. This could partly result from our data-set containing fewer classes (8 vs 11). Another factor could be our partially labelled test data, which features very few class boundary regions, however further testing with fully labelled data shows similar performance. It is possible that partially labelled training data could lead to a better performing classifier due to the lack of potentially confusing boundary pixels, although to fully test this we would need to compare these results to those obtained by training an identical network with a fully labelled version of the same data set, which is beyond the scope of this paper.

Data Set Size. To consider the effect the amount of training data used has on classification, we train networks using five different sized subsets of our training data, containing 140, 70, 35, 17 and 8 images, both with and without pre-training. Figure 7 compares results for three of these subsets, each trained for 10,000 iterations.

The effects of transfer learning are similar: for the first 1000 iterations, the benefits of pre-training are clear, however after just a few thousand more, both pre-trained and un-pre-trained networks have achieved close to their optimum performance. As training progresses, the pre-trained network consistently outperforms the non-pre-trained network by a small margin, which generally increases as the dataset size decreases: After

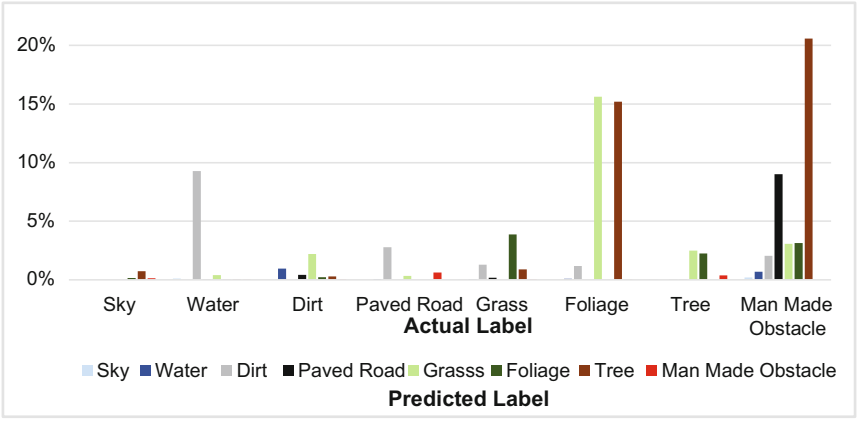

Fig. 8. Misclassified pixels, per class, as a percentage of the total number of pixels belonging to that class (correctly labelled pixels are not shown)

10,000 iterations with a dataset of 140 images, the accuracy of the pre-trained network is just 0.01 better than the un-pre-trained network, while with the dataset of 8 images, this margin increases to 0.09.

Per Class Results. We now discuss in more detail the results from the CNN trained for 10,000 iterations on the full data set after 30,000 iterations of pre-training. This is the network configuration that we would expect to typically perform best, with the highest amount of pre-training and largest data set, and it consistently achieves an accuracy of 0.91 against our partially labelled test data once it has passed 5000 iterations. Figure 8 shows the proportion of pixels belonging to each class that were given each possible incorrect label.

The most common misclassifications are between grass, foliage and trees, which is understandable given their visual similarities. Proportionally to class size, the largest is the 20.5 % of pixels containing Man Made Obstacles that are misclassified as Tree. This is likely because many of the man-made obstacles in the off-road environment, such as fences, posts and gates, are made of wood and so have a similar appearance to trees.

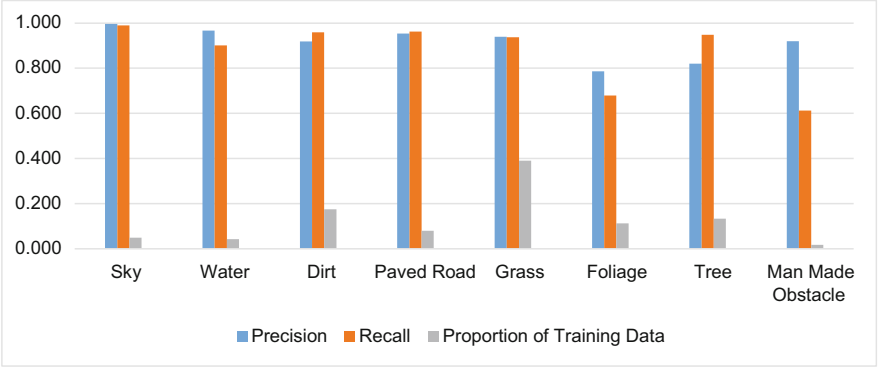

Fig. 9. Per class statistics for the CNN classifier. CNN was trained for 10,000 iterations on the full off-road data set after 30,000 iterations of pre-training. Testing was performed on the partially labelled testing set. As well as class precision and recall, we plot the number of pixels comprising each class within the training data as a proportion of the total number of labelled pixels in the set

Figure 9 plots the precision and recall of each class along with the proportion of the training data set that each class makes up. The foliage class performed worst, likely due to its visual similarity to both grass and trees, while sky gave the best results. Camera exposure was set to capture maximum detail at ground level, so in most instances the sky is much brighter than the rest of the scene, which combined with its lack of high frequency detail and consistent placement at the top of an image makes it easily distinguishable from other classes.

For the most part, classes that achieve high precision also achieve high recall, however man-made obstacle is an exception, with a very high precision (0.92) but

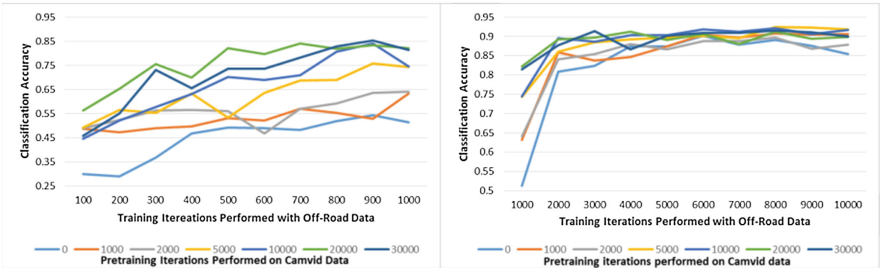

Fig. 10. Classification results using the fully labelled test set, comparing networks that have undergone different amounts of pre-training on the Camvid urban data set

lowest overall recall (0.613), meaning very few pixels are misclassified as man-made obstacle, while many pixels which should be labelled man-made obstacle are not. The fact that it is the class with fewest training samples (594,125 pixels) is likely to have played a part in this, as well as its visual similarity to trees, as discussed above.

There would appear to be some correlation between the frequency of a class within the data set and its recall, possibly because of the way the output is weighted towards classes that appear more often.

3.2 Fully Labelled Test Images

Currently we have only discussed the results obtained through testing the CNN classifier against partially labelled data, thus we also test it against a set of fully annotated images to demonstrate that it can achieve similar results.

Figure 10 show the results obtained, and demonstrates that testing with fully labelled images yields results very similar to those of the partially labelled set. The highest accuracy seen was with the network pre-trained for 5,000 iterations, with an accuracy of 0.924 after 8000 iterations of training with the full off-road data set.

Interestingly, the network snapshots that perform poorly on the partially labelled set (i.e. those that have not yet been through enough training iterations or have only been trained on a small data set) tend to perform worse on the fully labelled images. By contrast, those that perform well on the partially labelled data exhibit less deterioration, and in some cases even demonstrate an improvement in accuracy, when the fully labelled set is used. This would appear to suggest that a more comprehensively trained network performs much better in class boundary regions.

Another point of note is that with the partially labelled data set, a network that had undergone greater pre-training would almost always perform better, however, when testing with the fully labelled data set, the networks that have undergone 5000 and 10,000 pre-training iterations consistently outperform those with 20,000 and 30,000 iterations, although only by a very small margin, at the later stages of training. This could be because the networks that have undergone more pre-training begin to overfit to the data they were originally trained on. The fact that this only occurs when the fully

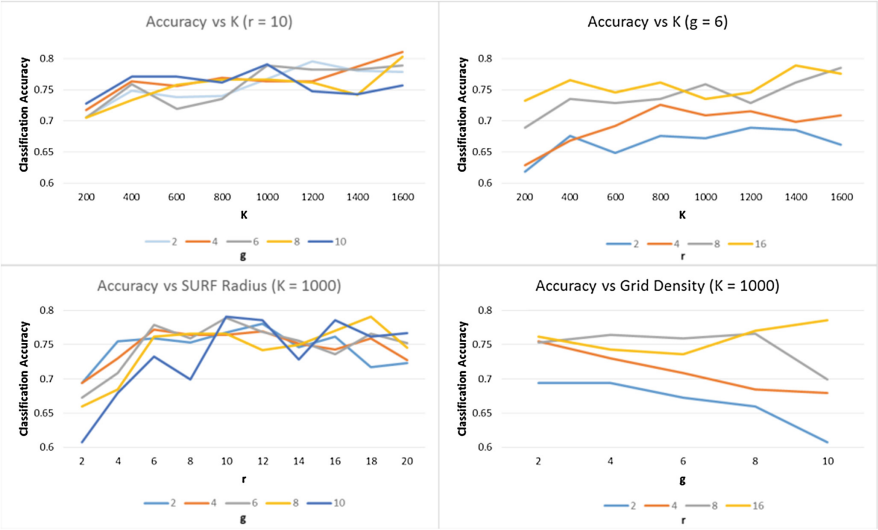

Fig. 11. Results from SVM classifier using various feature configurations. K represents the number of clusters used for bag-of-words encoding, g is the density of feature grid, i.e. number of pixels in both the x and y direction between feature points, and r is the radius, in pixels, of the area that each feature point takes account of when building its SURF descriptor

labelled data is used might suggest that this overfitting only has a noticeable effect when classifying class boundary regions, which are not present in the partially labelled data.

3.3 SVM

For Comparison, we test the SVM approach on its ability to classify segments from our off-road data set. The SVM parameters are automatically optimised through cross-validation, however we test several different configurations for the features that we pass into the classifier. The parameters that we alter are g, the number of pixels between feature points in our grid, r, the radius in pixels around each feature point that our descriptors take account of, and K, the number of clusters used to build our bag-of-words. Figure 11 shows several comparisons to demonstrate how performance is affected.

We would expect a decrease in g to improve results, as a greater amount of detail is being considered. This partly holds true in our results, although not consistently so. As r changes, we initially see a consistent improvement in results, which begins to tail off after a while. This is likely because with r set too restrictively, each feature point only has access to a limited region of local gradient information. By contrast, with r set too large, high frequency detail is lost as the descriptor is built from a greater number of pixels. The optimum value appears to be around $r = 10$. The general trend for K is that larger is better, but memory and time constraints make too large a value impractical.

The best result attained by the SVM was an accuracy of 0.813, using the parameters $g = 6$, $r = 12$, $K = 1400$.

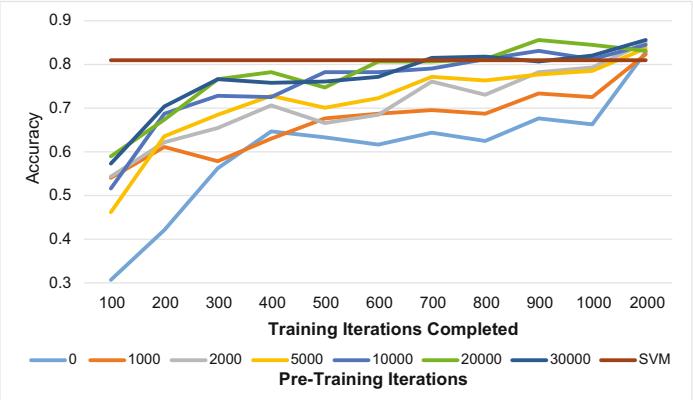

Fig. 12. Training progress of the CNN after different amounts of pre-training, measured as accuracy on the segment classification task for comparison to the SVM classifier

To properly compare SVM and CNN performance, we adapted our CNN classifier to label whole segments. This was done by winner-takes-all vote of pixel labels within the segment. Figure 12 shows the segment classification results as the CNN is trained after different amounts of pre-training. The CNNs pre-trained for 10,000 or more iterations all achieve results better than those of the SVM before 1000 iterations, and by 2000 iterations all, including the CNN that has undergone no pre-training, have surpassed the SVM. After further training, segment classification results are very similar to those for pixel classification, peaking at around 0.91, confirming that the CNN is significantly more effective at this classification task than the SVM.

4 Conclusions

This work demonstrates how an existing deep convolutional neural network classification and segmentation architecture can be adapted to a new task with minimal intervention. We have shown how quickly the network can learn to classify new kinds of images, and visualised CNN training performance by testing classification accuracy throughout the cycle, allowing us to compare networks as training progresses to show the effects that transfer learning and data-set size can have on performance.

Notably, we have demonstrated that pre-training is of limited utility when a large data set is used for a long training period. While pre-training was shown to improve performance when smaller data sets were used, results were still well below those observed with larger training data, even without pre-training, suggesting that pre-training is no substitute for an adequately sized data set. Pre-training was also shown to improve results early in the training cycle, although as training continues these effects diminish until a network with no pre-training will almost match the performance of a pre-trained one. In our testing, this happened as early as 5000 iterations, which represents just 5 h of training. However, our results have shown that networks that have undergone more pre-training tend to perform marginally better, even after many iterations of training,

however the results obtained from our fully labelled test set appear to show the opposite effect above 5000 iterations of pre-training, suggesting that there is a limit. With that in mind, it would appear the optimum configuration of CNN for this task, using the Segnet architecture [6] and trained on our full off-road data set, is around 10,000 iterations of pre-training followed by 10,000 iterations of training.

Our results show that such a CNN can outperform a SVM based classifier using dense gradient features by a significant margin, even after a limited amount of training.

References

1. Li, L.-J., Socher, R., Fei-Fei, L.: Towards total scene understanding: classification, annotation and segmentation in an automatic framework. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009. IEEE (2009)
2. Gupta, S., et al.: Indoor scene understanding with RGB-D images: bottom-up segmentation, object detection and semantic segmentation. Int. J. Comput. Vis. **112**(2), 133–149 (2015)
3. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
4. Tang, I., Breckon, T.P.: Automatic road environment classification. IEEE Trans. Intell. Transp. Syst. **12**(2), 476–484 (2011)
5. Alvarez, J.M., Gevers, T., LeCun, Y., Lopez, A.M.: Road scene segmentation from a single image. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 376–389. Springer, Heidelberg (2012)
6. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
7. Shin, H.-C., et al.: Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging **35**(5), 1285–1298 (2016)
8. Jansen, P., et al.: Colour based off-road environment and terrain type classification, Piscataway, NJ. IEEE (2005)
9. Manduchi, R., et al.: Obstacle detection and terrain classification for autonomous off-road navigation. Auton. Robots **18**(1), 81–102 (2005)
10. Chatfield, K., Zisserman, A.: VISOR: towards on-the-fly large-scale object category retrieval. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part II. LNCS, vol. 7725, pp. 432–446. Springer, Heidelberg (2013)
11. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
13. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: a high-definition ground truth database. Pattern Recogn. Lett. **30**(2), 88–97 (2009)
14. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
15. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: Proceedings of the Ninth IEEE International Conference on Computer Vision. IEEE (2003)