An Investigation of Transfer Learning for a Lifelog Dataset

Akanksha Rajpute, Tejal Nijai, and Graham Healy

School of Computing, Dublin City University, Ireland

Abstract. Achieving high image classification performance is often difficult when little training data is available, particularly when using deep learning approaches. Lifelog image classification is an example of this, where there can often be insufficient data available to directly train a deep learning model in an end-to-end manner. Moreover, the concepts (labels) in lifelog data sets tend not to align with the label outputs of pre-existing trained models, meaning these existing solutions are often unsuitable to be directly used without modification or training. Transfer learning has been proposed as a potential solution here, that is using the existing knowledge in a pre-trained deep learning model (for another image classification task) as a starting point for a new image classification task. Our image classification problem in this paper is about classifying daily activities from lifelog images. We evaluate two different types of transfer learning approaches to improve training time and performance in the presence of limited training data. In this paper, we outline a comparative study of two different transfer learning approaches: a) the application of traditional classifiers using features extracted by a pre-trained model and b) the fine-tuning of a pre-trained model. We benchmark these two different approaches for transfer learning using metrics for accuracy, recall, precision, and f1-Score to identify which approach performs best. For the LSC2018 dataset used in this study, we find that using a pre-trained VGG19 model as a feature extractor in combination with XGBoost to give the best performance in terms of accuracy.

Keywords: Transfer Learning · Lifelog · Feature Extraction · Fine-tuning· Pre-trained Models.

1 Introduction

Lifelogging represents a phenomenon whereby people can digitally record their own daily lives in varying amounts of detail, for a variety of purposes [10]. This could be for example to allow an individual to analyze their lifestyle or to keep a record of life experiences for memories [10]. Most lifelogging research has focused on visual lifelogging in order to capture life details (examples shown in Figure 1). Considering the high rate of image capture of lifelog wearable devices, over a million images could be captured in a year by a single user (assuming that a lifelog image is captured every 20 seconds over a 16-hour day). As the

sheer volume of images generated makes it infeasible for a user to manually label these images, automated methods are needed to extract rich metadata in the form of image concepts/labels in order to support processes like indexing, search and summarization [8]. These are vital components that a software system that uses lifelog data needs to have to support many different types of lifelog applications [10]. While many highly accurate image classification and concept detection models already exist that rely on deep learning approaches (e.g. [4,5]), the outputs of these pre-trained solutions often do not align with the image concepts lifelog software systems require. An ostensibly sensible solution here is to train new lifelog image concept detectors in the same way as these existing solutions, however, very often sufficient quantities of labelled training data are not available. This is particularly evident when considering the millions of labelled images that are commonly used to train popular image classification models [18]. One solution here is to use transfer learning, a process whereby the "knowledge" in a pre-trained model (trained on millions of labelled samples) for one task can be used as the basis for learning to solve a new problem where relatively little labelled training data may be available. Transfer learning is a prevalent technique in the deep learning community because it can both A) mitigate issues in training deep neural networks where relatively little data is available by infusing a base model with task-independent knowledge of visual primitives and concepts, and through this B) avoid long training times [2]. This is very efficient because most real-world problems typically do not have millions of labelled data samples. In this paper, we explore how transfer learning approaches can be used to overcome inherent limitations in a lifelog dataset that has relatively few labelled samples. There are various pre-trained models available which have performed well on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18], and thus are used in our work. For our investigation, we use the LSC2018 data [1], which is a dataset containing various pre-labelled lifelog images captured by an individual over a one month period. We compare two approaches of transfer learning using this labelled lifelog dataset. In the first approach, we use traditional machine learning approaches on features extracted using pre-trained deep convolutional network models. In the second approach we use the existing pre-trained models, and fine-tune the weights of the last layers of the model for our lifelog image classification problem. We seek to discover which existing pre-trained models (VGG16 [5] vs VGG19 [4]) provide the best accuracy for our custom classification problem defined on the lifelog dataset.

The structure of the paper is as follows: the related works are presented in Section II. Section III deals with the proposed methodology followed by the experimental results; Section IV describes results followed by discussion and conclusion in Section IV and Section VI, respectively.

2 Related Work

The use of CNNs (Convolutional Neural Networks) for image classification problems has become commonplace, as it gives excellent results compared to non-

CNN approaches in many applications domains [19]. We describe the current work in this area for different image classification problems, in order to highlight the most relevant techniques and methods available for improving activity classification of lifelog images. Previous work [16] has explored learning and transferring mid-level image representations using CNNs, in order to overcome problems related to limited labelled training data. In this work, image representations were learned using CNNs on a large-scale annotated dataset, where transfer learning was then used to accomplish other downstream visual recognition tasks. In the paper, the PASCAL VOC 2005 and 2012 datasets were used for an object class recognition problem using a transfer learning approach from a CNN model trained on ImageNet. By reusing the existing network and adapting new layers, classification and object recognition results were improved on this new task. Other work [13] has explored classifying images of tread patterns to help in providing useful knowledge for investigating criminal cases and coping with traffic accidents. The authors demonstrate using a CNN model (Alexnet [12]) as a feature extractor, and show increased efficiency through the application of transfer learning. In this work, features from single layers, including convolutional layers Conv3, Conv4 and fully connected layers fc6 and fc7 were compared in terms of performance as inputs for traditional machine learning methods to learn from. As the feature dimensionality for layers was large, Principle Component Analysis (PCA) was applied for dimensionality reduction. Additionally, dimensionality reduction was used in this way to eliminate interference components due to the noise. In their results they show that using PCA gives better accuracy than not using PCA. On these extracted features, a SVM Classifier was used to compare the performance with existing algorithms for tread pattern classification. In [7], transfer learning was used to classify fine-grained images. Fine-grained image classification was studied where instances of different classes share some common parts but have variation in shape and appearance. Classification of fine-grained images is a challenging task. For the experiment, the Stanford Dog dataset [11] was used to classify 120 breeds of dogs. A higher accuracy for recognition on the dog breed recognition problem was demonstrated using fine-tuning of VGG16 and InceptionV3 pre-trained models. The experiment was performed using four different approaches, namely: A) a simple sequential network, B) a simple sequential network with global average pooling, C) transfer learning with the Adam optimizer of the classifier, and D) transfer learning with Adamax training of the classifier. Transfer learning with Adamax training of classifier outperformed with an accuracy of 93% compared to the other approaches. This conveys the power of using fine-tuning on pre-trained models to classify hard-to-distinguish images. There are many pre-trained CNN models available that have been shown to perform well on image classification tasks. To decide on base models to use in our task, we refer to the following papers to models that have outperformed in different image classification problems. In [5], they examine using transfer learning on pre-trained models (Alexnet, VGG16, DenseNet201, GoogleLeNet, and ResNet) for flower classification. In this study, the CNN models were trained on the ImageNet dataset, and then were fine tuned on the flower's dataset [14].

The dataset consists of five classes, including chamomile, tulip, rose, sunflower and dandelion. The model design was discovered by trial and error, and by using adaptive methods. In this comparison, VGG16 performed well with 93.52% accuracy compared to Alexnet, DenseNet201, Google Net and ResNet. The authors in [4] propose a new approach for highly realistic computer-generated images detection by exploring inconsistencies in the region of the eyes. Such inconsistencies are captured by exploring the power of features extracted via the transfer learning approach with the VGG19 model. The VGG19 architecture was used for feature extraction, and on these features, a traditional machine learning classifier approach (SVM) was applied with an accuracy of 80%. [15] performed a comparative analysis on the fine-tuning of two pre-trained models (InceptionV3 and Xception) focusing on diabetic retinopathy screening. The authors describe approaches to fine-tuning pre-trained networks by studying different tuning parameters and their effect on the overall system performance due with respect to the application of diabetic retinopathy screening.

Another piece of work [3] which uses the InceptionV3 model performance for extreme learning machine (ELM) and fully connected layer was compared on overall training time and testing accuracy when applying transfer learning on pre-trained models such as VGG16, ResNet50, and InceptionV3. In this, they used the CIFAR10 and Fruit-360 datasets for their experiments. Other authors [17] have focused on the effect of well-known optimizers (Adam, SGD, and RMSProp) on CNN models namely, ResNet50 and InceptionV3. These optimizers were used to fine-tune the CNN models for 15 Epochs on a cat vs dog dataset generated by handpicking hundreds of images of cats and dogs from the Kaggle cat vs dog dataset. A lower learning rate (i.e. 0.001) was used in conjunction with categorical cross entropy. The experiment showed that the SGD optimizer outperformed the other two for ResNet50. An accuracy of nearly 97% was observed for ResNet50 with 500 training and 100 validation images.

From this related work, we identified VGG16 and VGG19 as suitable network architectures to explore transfer learning for classification of lifelog images for our experiments, since many previous studies have used these models with positive outcomes for conceptually similar types of experiments.

3 Methodology

3.1 Dataset Description

For our study, we use the LSC2018 dataset [1], which consists of 27 days (from 15 Aug 2016 to 10 Sep 2016) of data of an active lifelogger. The dataset is based on the NTCIR-14 Lifelog dataset [9]. The dataset includes 41,692 images captured by a wearable camera, multimedia files, and related metadata files (containing image labels) on a per-minute basis($\tilde{3}$ images per minute). For this research, we used the lifelog images and the corresponding metadata CSV file that contains labels for each of the images associated with the activities of the lifelogger. Some lifelog camera samples from the dataset are shown in Figure 1.

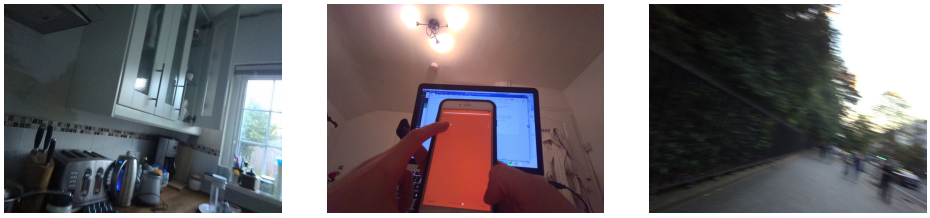

Fig. 1. Samples images from the LSC2018 Lifelog Image Dataset for labels: "at home" (leftmost), "using mobile" (center) and "walking" (rightmost).

3.2 Dataset Preprocessing

After filtering the images using the metadata CSV file (that contains the image labels), we found that 24,162 images of these images were annotated across 209 categories. We found that many of the categories were nearly similar e.g. *drinking water*, *drinking coffee*, *drinking tea* and *drinking beer* were all categories. In order to ensure there was sufficient labelled data to support our analysis, we combined labels for such similar and overlapping categories to make them into single categories based upon the dominant activity e.g. eating, drinking, walking. This first pass of merging categories resulted in 25 categories[1] shown in Table 1.

Issues were identified with some categories of images, for example some were blurred, some had very few sample images (e.g. less than 20), and some images semantically belonged to more than one category. For example, the center image in Figure 1 is labelled as "Using Mobile" in the dataset, but it has both a mobile and a laptop in the image.

In order to mitigate these issues, we sorted each category by the number of labelled images available (in descending order), and kept the top-10 categories for our analysis[2]. A breakdown of the number of labelled samples available per category is shown in Table 2.

3.3 Dataset Analysis and Approaches

The images dataset (after filtering) we used for experiments contained 19,527 images. Table 2 shows the breakdown of the distribution of images across the 10 categories. A stratified 60%, 20% and 20% split for training, validation and test set was used for all of our experiments. Analysis was carried out using deep learning virtual machine instances in the Google Cloud Platform along with using a local machine with a 4GB Nvidia 940MX graphic card. We investigated

[1] For example, categories such as "eating apple", "eating food", "eating curd", and "eating strawberry" were merged to one category to "eating". "In bedroom", "In living room" and "In a bedroom" were merged as "At home". Similarly, "Car", "Commuting to work in car", "Travelling in car" and "Travelling back from work in car" were merged as "Commuting in car".

[2] When 25 categories were used, the measured performance was poor as would be expected given many categories had insufficient images to be able to train with.

Table 1. List of 25 categories

Attending conference, presentation	At home	Brushing teeth	Casual conversation	Cleaning and washing
Commuting in car	Drinking	Eating	Gardening	In airplane
Organising things	Other	Printing	Reading paper	Relaxing
Retail shopping and purchasing	Travelling	Using desktop computer	Using laptop	Using mobile
Using tablet	Walking	Watching TV	Work meeting	Writing

Table 2. Ranked number of labelled images per category (across 10 categories)

Category	**Number of Samples**
Using laptop	4440
Relaxing	3055
Walking	2467
Commuting in car	2181
Using desktop computer	1934
Work meeting	1511
Casual Conversation	1180
In airplane	1028
Watching TV	882
Eating	849

2 approaches to perform transfer learning as we wanted to compare the performance of each to identify which approach performed best for the classification of lifelog activities. For this, we use a pre-trained VGG19 model[3].

We followed similar feature extraction and fine-tuning methods as described in the papers in our related work section (detailed below). Moreover, instead of considering only accuracy as an evaluation metric, we have considered precision, recall, and F1-score also because accuracy is not always an informative metric when there are imbalances in a labelled dataset i.e. if any label in the dataset is disproportionately more frequent then a classifier could learn to blindly predict this label and still have a high accuracy. Due to this reason, other metrics like precision, recall, and F1 score are calculated and considered. The precision measures give insight about the false positive rate. The recall indicates how many true positives (correctly predicted labels) are found from all the positive samples in the test set. F1-score is the harmonic mean of precision and recall, and a useful metric for imbalanced dataset evaluation. In our analysis, we concluded to three highly performing classifiers in feature extraction approach. In addition to that, we have also adopted three different ways of fine-tuning the transfer learning process by only training certain layers and freezing the others.

[3] The VGG19 model is available from https://keras.io/api/applications/

Approach 1: Using CNNs as Feature Extractors
For *Approach 1* we used a pre-trained VGG19 model as a feature extractor in combination with traditional machine learning classifiers, as this is one of the most common ways to do transfer learning [6]. In this process, features are extracted from images using pre-trained models, where the activations at particular layers of the model are used as feature vectors for other machine learning methods (e.g. SVM). In our work, features were extracted on a layer-by-layer basis (starting from the penultimate layer to the output layer) from CNN models in order to examine the efficiency of features (activations) at that layer.

Using the extracted image-related features on layers of the pre-trained CNN, we benchmarked several machine learning classifiers i.e. SVM, Random Forest, Naive Bayes, XGBoost, and Bayesian ridge regression. We used principal component analysis (PCA) to reduce the dimensionality of the extracted features, selecting the number of components that would retain 95% of the variance of the original data. We used PCA because features extracted at fully-connected layers were prohibitively large to be sensibly used with our classifiers. For VGG16 and VGG19 models, the last three layers define the classifier layer i.e fully connected (fc1), fully connected (fc2) and predictions (dense) layer. We also explored hyper-parameter tuning of the classifiers to improve the performance. For hyperparameter tuning, we used a grid search cross-validation method to find the optimal parameters for classifiers (using the validation set). For example, in the case of SVM, we explored three hyper-parameters i.e values of C, gamma and kernel type. For random forests, we explored number of estimators, max features,max depth and criterion to maximize the performance. We explored hyperparameters such as max depth, number of estimators and learning rate with the XGBoost classifier. The detailed results for the classifiers' outcomes for VGG19 are in Table 3.

Approach 2: Fine Tuning
For *Approach 2*, we used fine-tuning (with a pre-trained VGG19 model) as a transfer learning technique. In this approach, we considered three different ways to do fine-tuning. For *Method 1*, we replaced the last layer of the model and only trained this layer freezing all others i.e. predictions (dense) with a dense softmax layer as output. For *Method 2*, we froze all the layers of the models (VGG16 and VGG19) except the last eight layers and trained their weights. We selected the last 8 layers as trainable because we wanted not only classifier layers to be updated in training but also the convolutional layers. For *Method 3*, we replaced the top layers of the model (i.e. the dense and classifier layers) and replaced them with a dense fully connected layer (with ReLu activation function) and a softmax output layer. We used a SGD optimizer (no minibatching) with a learning rate of 0.0001. For all methods we used a sparse_categorical_cross-entropy loss function. To get a good fit for each model, each model was trained over a number of epochs until a good fit model was achieved as shown in Figure 2.

Table 3. Approach 1 - Feature Extraction Score Table (in percent) of VGG19 for 10 categories

VGG19					
Layer Name (where features were extracted) and respective evalution metrics	**SVM**	**Random Forest**	**XGBoost**	**Naive Bayes**	**Bayesian Ridge Regression**
Accuracy					
Predictions(Dense)	**82.27**	**81.80**	**85.83**	52.15	71.40
Fully Connected 2(fc2)	**86.67**	**83.24**	**88.60**	66.91	82.03
Fully Connected 1(fc1)	**87.08**	**83.55**	**89.60**	71.48	76.78
Flatten	**79.18**	**74.20**	**80.03**	72.30	68.47
Precision					
Predictions(Dense)	**82.82**	**76.64**	**85.36**	52.78	69.91
Fully Connected 2(fc2)	**87.34**	**85.13**	**88.76**	67.14	77.11
Fully Connected 1(fc1)	**88.48**	**85.71**	**89.90**	70.59	84.04
Flatten	**72.13**	**71.77**	**75.55**	44.84	57.71
Recall					
Predictions(Dense)	**78.82**	**76.20**	**83.75**	60.50	68.07
Fully Connected 2(fc2)	**84.85**	**80.48**	**87.38**	71.08	75.22
Fully Connected 1(fc1)	**85.03**	**80.18**	**88.28**	73.26	75.02
Flatten	**64.09**	**65.14**	**66.19**	44.66	45.33
F1 Score					
Predictions(Dense)	**80.79**	**76.43**	**84.56**	56.37	68.98
Fully Connected 2(fc2)	**86.10**	**82.72**	**88.07**	69.05	76.15
Fully Connected 1(fc1)	**86.78**	**82.83**	**89.09**	71.90	79.27
Flatten	**67.86**	**68.31**	**70.51**	44.74	50.77

As per the learning curves shown in Figure 2, it can be seen that each model began to show signs of overfitting at a different number of epochs, indicating that it was possible to overtrain and in some cases this was detrimental to performance. For *Method 1* (replacing the last layer) this occurred at approximately 17 epochs where accuracy on the validation set began to decrease while training set accuracy continued to increase. For *Method 2*, we can see the model learns for the initial 2-3 epochs after which performance on the validation set stabilises while the training set accuracy continues to increase. For *Method 3*, we can see the validation set accuracy only marginally improves after an initial learning period up to 3 epochs.

Table 4 shows the *Approach 2* results for VGG19 [4]. We evaluated performance on the test set using accuracy, precision, recall and f1-score. Here, the importance of F1-score is very high because our dataset has an uneven class

[4] From our analysis, VGG16 performed marginally worse than VGG19 and for this reason we don't include the results

distribution. F1-score gives us the harmonic mean of precision and recall. So, for a model to be accurate F1-score should have a high value.

Table 4. Approach 2 - Fine-Tuning Score Table (in percent) for 10 categories of VGG19

VGG19				
Methods	**Accuracy**	**Precision**	**Recall**	**F1-Score**
M1: Replacing Last Prediction Layer	77.63	77.15	74.37	75.73
M2: Freezing layers except last 8 layers	86.68	85.86	85.28	85.54
M3: Replacing classifier layers with other layers	81.19	80.70	78.93	79.56

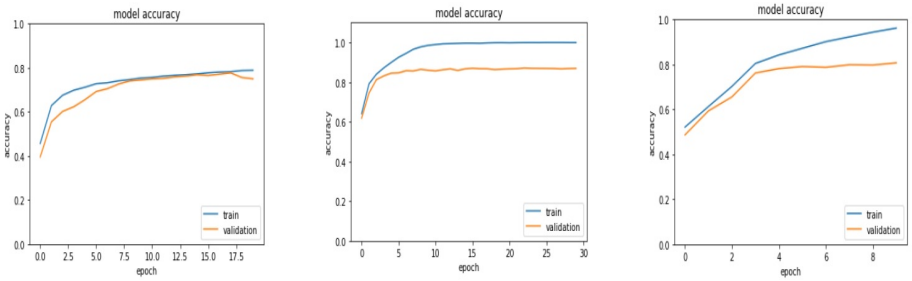

Fig. 2. Training Curves of VGG19 for Three methods: Method1(leftmost), Method2(center), Method3(rightmost)

Here, in Figure 2 we can see that the plot of method 1 where only last layer was removed seems to be a good fit and the other methods are overfit or underfit. Whereas method 2 performed better.

4 Results

Comparing model performance with different classifiers in *approach 1*, VGG19 and XGBoost emerged as a better combination with an average accuracy of all classifier layers outcome as 86.015%, precision of 84.89% , recall of 81.40% and f1-score 83.05%. In *approach 2*, VGG19 emerged as the better model too on all three applied methods with an accuracy of 86.68%, precision of 85.86%, recall of 85.28% and f1-Score as 85.54% when we kept only last 8 layers trainable (method 2).

Fig. 3. Screenshots of Incorrect Predictions

When we compare *approach 1* and *approach 2*, we can see that *approach 1* gave a higher accuracy (with XGBoost). Also, *approach 1*-Feature Extraction and fitting a classifier is the most common approach for transfer learning[5].

5 Discussion

In this supervised machine learning classification study, we found that labelling of the images is very crucial. As a human being, we often identify one aspect of the image while labelling, that might not be correct in machine's and model's perspective i.e. label noise in training and evaluation. For example, there are some images in the categories "Work meeting" and "Reading paper" that contain both papers and a laptop(s) in the images. However, these images are different from the perspective of a lifelogger but could create confusion during model training and prediction. In effect, some images belong to multiple classes e.g. in Figure 3 the person has a phone and is in an airplane so the image could belong to "Using mobile" or "In airplane", but it was labelled as "Using mobile" but predicted as "In airplane". This is an clear example of two overlapping categories. Similarly, the second image is predicted as "Using desktop computer" when it is actually an image of a home. Such mislabelled samples in the training, validation and testing sets could be introducing (label-)noise into our training and evaluation procedure, resulting in sub-optimal results. In future work, we will examine how to mitigate such issues when dealing with real-world datasets.

6 Conclusion

In this paper, we conducted our research using the top-10 labelled categories of the LSC2018 dataset i.e. had the most numbers of samples per category. We explored two transfer learning techniques using a pre-trained VGG19 model, namely feature extraction and fine-tuning. In turn we evaluated the performance of these two different approaches using precision, recall, accuracy and f1-score. For our first approach, we benchmarked a battery of classifiers using the VGG19 model (across a number of layers) as a feature extractor e.g. XGBOOST, SVM

[5] In our experiments, VGG19 emerged as the better performing model for transfer learning in both the approaches.

and random forest. Here we find that using XGBoost using the fully connected layer 1 features of the pre-trained VGG19 model produced the best results. In our second approach (fine-tuning), we used three distinct approaches, where we found only training the last 8 layers produced the best accuracy for this approach. Although we find that fine tuning is an effective approach (with 86.68% accuracy for method 2), the accuracies for all methods in this approach were surpassed by simply using the pre-trained VGG19 model as a feature extractor in combination with XGBoost model (89.6% accuracy) .

Future studies will assess the impact of dataset labelling and methods to reduce labelling noise. Similarly, in future work we will examine the impact of data availability to see if additional data might improve the fine-tuning approach. We conclude that in instances where there is little labelled data that using a pre-trained model as a feature extractor is a suitable approach to classify images of daily activities present in a Lifelog.

References

1. Lsc 2019 @ icmr 2019, http://lsc.dcu.ie/2019/data/index.html
2. Alom, Z., Taha, T.M., Yakopcic, C., Westberg, S., Nasrin, S., Asari, V.K.: Comprehensive Survey on Deep Learning Approaches (2017), https://arxiv.org/pdf/1803.01164.pdf
3. Alshalali, T., Josyula, D.: Fine-tuning of pre-trained deep learning models with extreme learning machine. Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018 pp. 469–473 (2018). https://doi.org/10.1109/CSCI46756.2018.00096
4. Carvalho, T., De Rezende, E.R., Alves, M.T., Balieiro, F.K., Sovat, R.B.: Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN. Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017 **2017-Decem**, 866–870 (2017). https://doi.org/10.1109/ICMLA.2017.00-47
5. Cengil, E., Cinar, A.: Multiple classification of flower images using transfer learning. 2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019 (2019). https://doi.org/10.1109/IDAP.2019.8875953
6. Erfani, S.M., Rajasegarar, S., Karunasekera, S., Leckie, C.: High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. Pattern Recognition **58**, 121–134 (2016)
7. Golodov, V.A., Dubrovina, M.S., Paziy, A.S.: Transfer Learning Approach to Fine-Grained Image Classification. Proceedings - 2019 International Russian Automation Conference, RusAutoCon 2019 pp. 1–5 (2019). https://doi.org/10.1109/RUSAUTOCON.2019.8867653
8. Gurrin, C., Joho, H., Hopfgartner, F., Zhou, L., Albatal, R., Healy, G., Nguyen, D.T.D.: Experiments in lifelog organisation and retrieval at ntcir. In: Evaluating Information Retrieval and Access Tasks, pp. 187–203. Springer (2020)
9. Gurrin, C., Joho, H., Hopfgartner, F., Zhou, L., Ninh, V.T., Le, T.K., Albatal, R., Dang-Nguyen, D.T., Healy, G.: Overview of the ntcir-14 lifelog-3 task. In: Proceedings of the 14th NTCIR conference. pp. 14–26. NII (2019)
10. Gurrin, C., Smeaton, A.F., Doherty, A.R.: LifeLogging: Personal big data. Foundations and Trends in Information Retrieval **8**(1), 1–125 (2014). https://doi.org/10.1561/1500000033

11. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. Proc. IEEE Conf. Comput. Vision and Pattern Recognition (2011)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
13. Liu, Y., Zhang, S., Wang, F., Ling, N.: Tread Pattern Image Classification using Convolutional Neural Network Based on Transfer Learning. IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation **2018-October**, 300–305 (2018). https://doi.org/10.1109/SiPS.2018.8598400
14. Mamaev, A.: Flowers recognition (Jun 2018), https://www.kaggle.com/alxmamaev/flowers-recognition
15. Mohammadian, S., Karsaz, A., Roshan, Y.M.: Comparative Study of Fine-Tuning of Pre-Trained Convolutional Neural Networks for Diabetic Retinopathy Screening. 2017 24th Iranian Conference on Biomedical Engineering and 2017 2nd International Iranian Conference on Biomedical Engineering, ICBME 2017 (December) (2018). https://doi.org/10.1109/ICBME.2017.8430269
16. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 1717–1724 (2014). https://doi.org/10.1109/CVPR.2014.222
17. Poojary, R., Pai, A.: Comparative Study of Model Optimization Techniques in Fine-Tuned CNN Models. 2019 International Conference on Electrical and Computing Technologies and Applications, ICECTA 2019 pp. 1–4 (2019). https://doi.org/10.1109/ICECTA48151.2019.8959681
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
19. Zhang, Q., Zhang, M., Chen, T., Sun, Z., Ma, Y., Yu, B.: Recent advances in convolutional neural network acceleration. Neurocomputing **323**, 37–51 (2019). https://doi.org/10.1016/j.neucom.2018.09.038