Human and Machine Recognition of Transportation Modes from Body-Worn Camera Images

Sebastien Richoz
Wearable Technologies Lab
University of Sussex
Brighton, United Kingdom
sr569@sussex.ac.uk

Mathias Ciliberto
Wearable Technologies Lab
University of Sussex
Brighton, United Kingdom
m.ciliberto@sussex.ac.uk

Lin Wang
Centre for Intelligent Sensing
Queen Mary University London
London, United Kingdom
lin.wang@qmul.ac.uk

Phil Birch
Engineering and Informatics
University of Sussex
Brighton, United Kingdom
p.m.birch@sussex.ac.uk

Hristijan Gjoreski
Faculty of Electrical Engineering and
Information Technologies
Ss. Cyril and Methodius University
Skopje, Macedonia
hristijang@feit.ukim.edu.mk

Andres Perez-Uribe
Institute for Information and
Communication Technologies
University of Applied Sciences
Yverdon, Switzerland
Andres.Perez-uribe@heig-vd.ch

Daniel Roggen
Wearable Technologies Lab
University of Sussex
Brighton, United Kingdom
daniel.roggen@ieee.org

***Abstract*—Computer vision techniques applied on images opportunistically captured from body-worn cameras or mobile phones offer tremendous potential for vision-based context awareness. In this paper, we evaluate the potential to recognise the modes of locomotion and transportation of mobile users, by analysing single images captured by body-worn cameras. We evaluate this with the publicly available Sussex-Huawei Locomotion and Transportation Dataset, which includes 8 transportation and locomotion modes performed over 7 months by 3 users.**

We present a baseline performance obtained through crowd sourcing using Amazon Mechanical Turk. Humans infered the correct modes of transportations from images with an F1-score of 52%. The performance obtained by five state-of-the-art Deep Neural Networks (VGG16, VGG19, ResNet50, MobileNet and DenseNet169) on the same task was always above 71.3% F1-score. We characterise the effect of partitioning the training data to fine-tune different number of blocks of the deep networks and provide recommendations for mobile implementations.

***Index Terms*—Activity recognition, Body-worn camera, Computer Vision, Deep learning, Crowd sourcing, Mechanical Turk.**

I. Introduction

The mode of transportation delivers an important contextual information about users. Modes of transportation include walking, cycling, taking a bus, driving a car, etc. The knowledge of the transportation mode assists context-aware applications such as localization, activity and health monitoring, parking spot detection, or content delivery optimization [1]–[3].

A user often carries a wearable device (e.g. smartphone, smartwatch, wearable camera) during travel, which is embedded with multimodal sensors including motion sensors, GPS (global positioning system), microphone and camera. There have been many studies on analysing the mode of transportation from the multimodal data captured by the smartphone sensors with machine learning techniques [4]–[10]. Motion and GPS sensors are widely used as they directly carry the orientation and motion information of the mobile device and the speed and trajectory of the user. The state-of-the-art in motion-based transportation recognition performance was established in the SHL recognition challenge 2018 [11], [12], which reveals that approaches based on motion sensors still struggle distinguishing between distinct transportation modes of similar kind: for example between train and subway (rail transport) or between bus and car transport (road transport).

Vision is an important modality that is available in wearable devices. There has been an increasing number of work that use wearable camera for life-logging, i.e. to record the surrounding environment and the daily life activities of people [13], [14]. Computer vision has progressed significantly with the introduction of deep-learning techniques. However, using images or videos captured from wearable devices to automatically recognize the user's context is still in its infancy. To the best of our knowledge, there has been no published work that reports recognizing the user's mode of transportation and locomotion from images captured by a wearable camera. One of the main reason for this is the limited availability of transportation dataset with vision data available. In an exhaustive review of available datasets, only the one which we use in this article has vision data [15]. Furthermore, visual information analysis with deep learning is computationally demanding and is only recently becoming possible in mobile devices [16].

In this paper we present the first work to evaluate whether vision can be used to detect the transportation mode of the user effectively. We use the state-of-the-art Sussex-Huawei locomotion-transportation (SHL) dataset [15], [17] that contains 86075 images collected by a body-worn camera over 7 months by 3 users who engaged in eight transportation activities II). We first present a baseline performance obtained by human visual inspection and classification of the images,

This work was supported by the HUAWEI Technologies within the project "Activity Sensing Technologies for Mobile Users". We thank NVidia for their GPU donation.

using Amazon Mechanical Turk (sec. III). We then present five vision-based deep-learning pipelines to recognize these eight transportation activities (sec. IV) and compare the performance to that obtained by human visual classifications. We further discuss technical aspects to improve the recognition performance, including identifying the best number of neurons to set in the final layers and the best partitioning of the available data train or fine-tune the weights at different depth of the network (sec. IV). Finally, we conclude comparing the findings of vision-based recognition obtained by deep learning to the one obtained by motion sensors, and emphasize the complementarity of the approach. We also provide recommendations for the architectures to use in mobile settings (sec.V). Accuracy, precision, recall and F1-score are used to interpret performance measures [18].

II. Dataset

The Sussex-Huawei Locomotion-Transportation (SHL) dataset is one of the biggest multimodal dataset for transportation and locomotion mode recognition from mobile devices [15]. The dataset was recorded over 7 months by 3 participants engaging in 8 different transportation modes: Still, Walk, Run, Bike, Car, Bus, Train, Subway (Sub.). The duration of the dataset is 2812 hours, corresponding to a travel distance of 17562 km in the south-east of UK. The data was recorded using 4 smartphones placed at different body positions and one body-worn unstabilised camera on the chest. As a result, the dataset contains 16 sensor modalities including motion, GPS, sound and image. In this paper, we only use the image data for analysing the mode of transportation.

The images were captured by the camera every 30 seconds with a size of 1024 × 576 (resized to 224 × 224 before processing). The dataset contains 86075 images, with their distribution among the 3 users and 8 classes shown in Fig. 1. For computational reasons, we use a subset of the complete set of images. We extract the same amount of images of each class, randomly among the users, which yields 14600 images, which constitutes 17% of the whole dataset. We randomly split this subset into train (70%), validation (18%) and test (12%) sets, while maintaining balance between classes.

Fig. 2 shows three exemplary images per activity class. Since the camera is unstabilised the rotation, blur, lighting conditions and the recording quality vary significantly among images, which makes the classification task challenging. Visual inspection shows that some classes may be easier to distinguish than others. For instance, the cycling activity can be deduced from the handle bar. However some other activities appear much more challenging to recognise, such as distinguishing walking from running.

III. Human performance baseline

We first sought to evaluate the performance of human identification of the transportation class from the pictures in the test set. This allows us to identify a human performance baseline to compare machine learning to. In order to gather a large amount of human-made classifications, we used Amazon Mechanical Turk (MTurk) [19]. This is a service that lets humans - called Requesters - create tasks to be fulfilled by other humans - called Workers - against a financial compensation. Such a task is called a Human Intelligence Task (HIT).

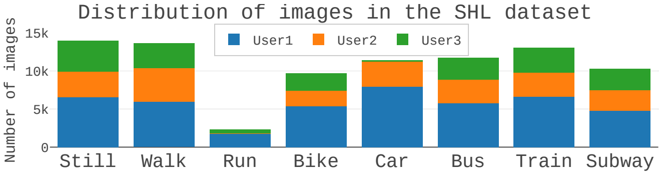


Fig. 1: Distribution of the images in the SHL dataset among the 8 transportation modes. We sample randomly 1825 images from each class to obtain a balanced dataset.

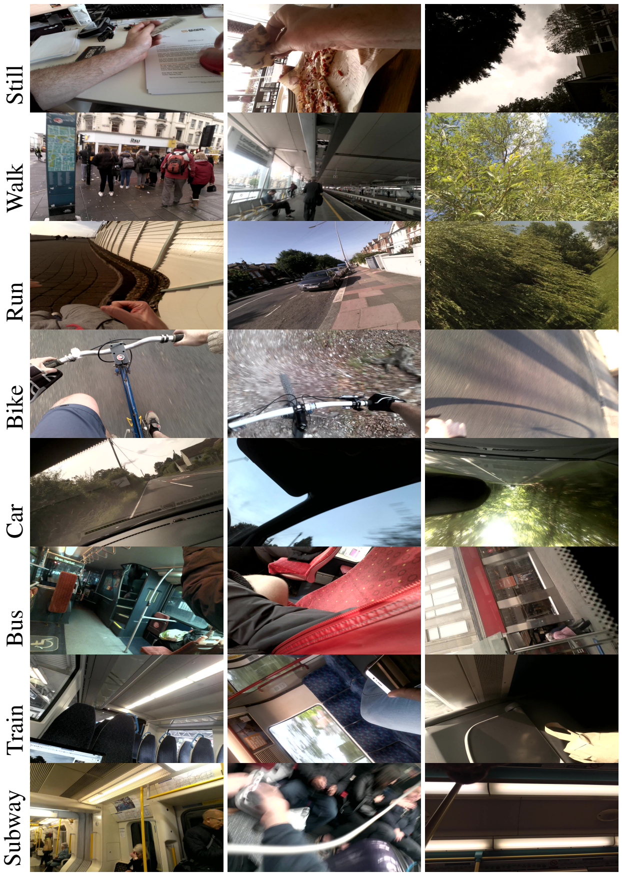
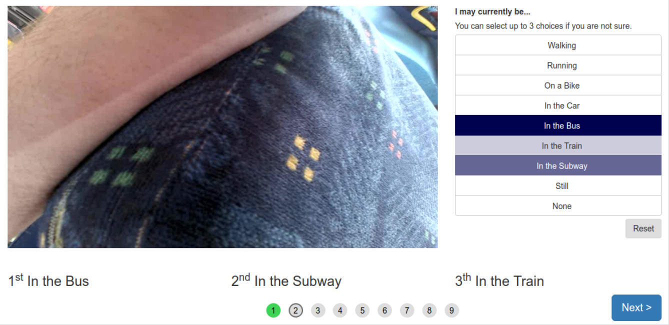


Fig. 2: Some pictures of SHL taken by the body-worn camera. Sometimes the locomotion mode is obvious thanks to distinguishable elements (handle bar, seats) and sometimes it is more challenging (Walk v.s. Run or looking through the window from a Train or a Bus).

A. *Creation and publication of the task*

The layout presented to Workers must be self-explanatory enough so that they can quickly fulfil their task. As a result and for budget optimisation, a Worker annotates 9 images at a time. For each image they can specify 1, 2 or 3 locomotion modes. For measuring the performance we used only the first



I may currently be...

You can select up to 3 choices if you are not sure.

1st In the Bus 2nd In the Subway 3th In the Train

Fig. 3: The task completed by each Worker. A Worker can select up to three locomotion modes to annotate the image (s)he is currently seeing. The green dots help to understand how many images are left before submitting the task.

given answer (i.e. the modes of locomotion they are the most confident to have identified). Fig. 3 shows the layout presented to the Workers. Additional instructions with examples helped them complete the task.

A single HIT consists in annotating 9 images and is rewarded \$0.12. Each image is guaranteed to be annotated by 3 different Workers. The 1800 images from our test set represent altogether 600 HITs, 5400 images and costs \$72. To this price we add \$12 of MTurk fees (\$0.02 per HIT) for a total of \$84. We didn't ask for Workers with master qualifications (Workers with approved experience on completing MTurk tasks) because we wanted a raw baseline as a start.

B. *Results and quality filters*

The task was completed in 1 hour and 25 minutes implicating 104 unique workers, annotating between 9 to 387 images each, with an average of 58.4 images annotated per Worker. From the 24.3% of images we were able to track [1], 54.3% were answered from India, 23.2% from Italy, 15.8% from USA, 5.3% from Ireland and less than 2% from Spain and Canada.

Some Workers may complete the HITs without taking care of the task. They could do it the fastest way possible to make the most money. Considering this, we applied a quality filter to remove all images annotated under a threshold called 'cut-off' time, which is between the minimum (640ms) and the first inter-quartile (2863ms) of all the annotations times (Fig. 5).

Then, we remove the images annotated by humans as 'None', which means they cannot recognise the mode of transportation, otherwise we would have to add an additional 'Null' class for the computer-based algorithms.

Fig. 4 shows the evolution of the F1-scores in function of the response cut-off threshold after applying the two above filters. The maximum F1-score is reached with a cut-off time of 1580ms. This represents an image loss of 1.24% (67 images) due to the cut-off time and 10.96% (592 images) due to the removal of 'None' annotations. Fig. 5 shows the times Workers took to annotate each locomotion mode. Additional filtering has to be done carefully because we could add bias by filtering potentially non-cheating Workers. We therefore keep our filter permissive enough to remove most of the presumed cheaters while keeping most of the presumed conscientious Workers.

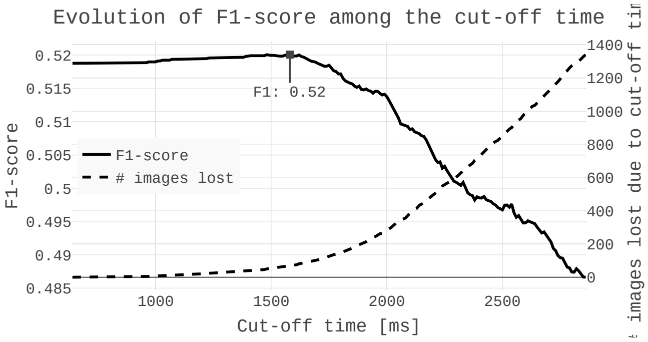


Fig. 4: The highest F1-score is reached when we remove annotations answered below 1580ms which we think is a good filter value to detect cheating. Above 1580ms the F1-score decrease significantly as we may lose correctly annotated images answered rapidly.

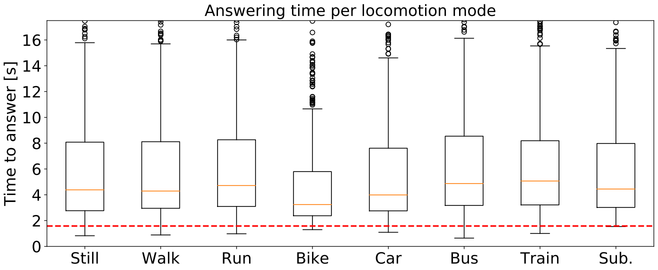


Fig. 5: The threshold for filtering cheaters is 1580ms (red line on the plot). This line is between the minimum time to answer and the 1st inter-quartile of all answering times.

For the human evaluation, we end up with a test set reduced to 4741 images annotated with one of the 8 transportation modes. We keep the duplicate images because 2 different Workers may annotate the same image differently as opposed to a computer-based classifier. The F1-score obtained before applying the quality filters is 51.88% and 52.01% after filtering. Fig. 6 shows the confusion matrix after filtering. This defines a baseline for the computer-based algorithms. However, we keep the full test set (1800 unique images) for the evaluation of the computer-based algorithms.

IV. Machine learning pipeline

Five state-of-the-art Convolutional Neural Networks (CNNs) were selected according to their size, number of weights and evaluation speed. They were all pre-trained on ImageNet challenge 2015 [20] (1.2M images and 1000 classes): VGG16 [21], VGG19 [21], ResNet50 (RN50) [22], MobileNet (MN) [23], DenseNet169 (DN169) [24]. Their characteristics are provided in Table I. They receive as input a 224 x 224 pixels picture and output a continuous value for each of the 1000 classes of ImageNet with a confidence

[1] We don't know the origin of 100% of the Workers because we could retrieve the IP address of only 24.3% of them.

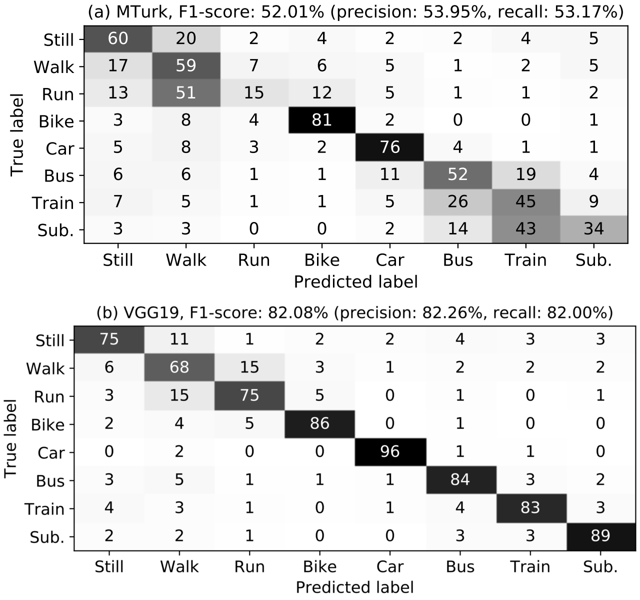


Fig. 6: Confusion matrix of the human performance (a) and the best computer-based algorithm (b). VGG19 outperforms the human evaluation in distinguishing Run from Walk as well as the vehicle transportation modes (Bus, Train, Subway).

score. These models were not designed to classify first person photos of transportation modes, we will use transfer learning and train additional layers at the end of the network for our purpose on the SHL dataset. The implementation is made in Python 3.6 with Keras 2.2.4 using the GPU version of tensorflow 1.12, CUDA 9.0 and cuDNN 7.3.1. For computation performance, we used four machines with NVIDIA Geforce GTX graphic cards: 1x TITAN XP, 2x 1080 Ti and 1x 1070.

A. Description of the Pipeline

Each classifier takes as input a 224x224 pixels image, processes it through B blocks and outputs a confidence value for each of the 1000 ImageNet classes. Each block B is made of several convolutional (Conv), pooling (Pool) and/or reducing (Redu) layers depending on the complexity of the network architecture. The blocks of each model are sequentially organised in the pipeline so that block B_{i+1} takes as input the output of B_i.

In our experiments, we adapted the original pipelines proposed for ImageNet [21], [24] to be applicable to the SHL dataset. First, the output layer is replaced with a global average pooling (GAP) layer to reduce the number of parameters. Second, a fully-connected (FC) layer $FC_1(N)$ is linked with the 'ReLu' activation function (N is the number of neurons). Third, we add a final FC layer $FC_2(8)$ with Softmax activation to output each of the 8 classes alongside a confidence score. Fig. 7 illustrates the resulting pipeline after our adaptation.

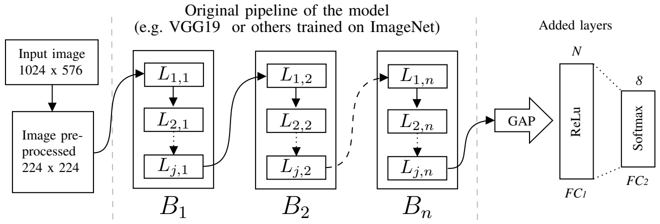


Fig. 7: Machine learning pipeline. Each model has B blocks which contains j layers $L_{j,n}$ according the model's architecture (see Tab. I).

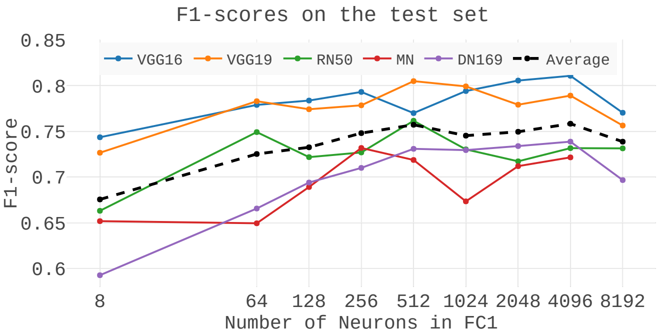


Fig. 8: Influence of the number of neurons in the fully connected layer. VGG16 and DenseNet169 are more sensitive to the number of neurons. The highest performance is globally reached with 512 neurons. We removed the surprising performance with MobileNet and 8192 neurons as it appears to outline a computational artefact in the training process.

B. Experiment 1: Optimizing the number of output neurons

We investigate the number of neurons N in $FC_1(N)$ which maximises the F1-score on the test set. Each classifier is trained (10240 images) on 100 epochs , validated (2560 images) and tested (1800 images). A stopping criterion stops the training prematurely if the validation's accuracy doesn't improve by at least 0.4% over 5 epochs. The training is done only on the last layers GAP, $FC_1(N)$ and $FC_2(8)$ where the weights are randomly initialized. The optimizer used is 'RMSProp' with a learning rate of 0.001 and the 'Categorical Crossentropy' loss function. The rest of the network was already pre-trained on ImageNet. The results are shown on Fig. 8. The F1-Score on the test set increases from 8 to 256 neurons and decreases after 4096 neurons. When averaging the F1-scores accross all classifiers, the maximum ones are obtained with 4096 (F1 75.8%) and 512 (F1 75.7%) neurons. We define our optimal number of neurons as 512 to save computational time in the following experiments.

C. Experiment 2: Fine-tuning

With the number of FC_1 neurons ($N = 512$) identified in the previous section, we explore here to which extent 'fine-tuning' the pre-trained classifier weights to specialise them to the SHL dataset is beneficial. We will use a part of the SHL dataset to fine-tune the last n blocks of the classifiers.

	ImageNet Accuracy	SHL (train only) Accuracy	SHL (train+fine-tune) Accuracy	Size	Parameters	Depth	Layers	Blocks	Layers/block (average)
VGG16	0.713	0.791 (0.012)	0.814 (0.009)	528 MB	138,357,544	23	19	5	3-4 (3.6)
VGG19	0.713	0.789 (0.008)	**0.821 (0.002)**	549 MB	143,667,240	26	22	5	3-5 (4.2)
MobileNet	0.704	0.713 (0.017)	0.770 (0.004)	16 MB	4,253,864	88	87	13	6-7 (6.3)
ResNet50	0.749	0.739 (0.018)	0.791 (0.005)	99 MB	25,636,712	168	168	16	10-12 (10.5)
DenseNet169	0.762	0.731 (0.022)	0.762 (0.002)	57 MB	14,307,880	169	595	82	7-7 (7)

TABLE I: Performance and characteristics of the CNNs trained and evaluated on ImageNet and SHL. For the SHL(train only), the results reported are trained with 512 neurons on 30 epochs. For SHL(train+fine-tune), the results reported are fine-tuned using 100% of the fine-tune set and the highest number of blocks, during 30 epochs. MobileNet is well suited for mobile implementation due to its low size. VGG16 and VGG19 have the largest size but their architecture remains simple due to the low number of layers. ResNet50 and DenseNet169 are smaller in size but have more layers and are deeper.

The process is made in two steps. First, in the fine-tune step (step 1), we freeze the first $B - n$ blocks while keeping the last n unfrozen as well as the last layers (GAP, $FC_1(512)$, $FC_1(8)$). Second, in the train step, (step 2) we freeze all the B blocks while keeping the last layers unfrozen.

The train set is split randomly into a fine-tune and train set according to a percentage P (e.g. $P = 40\%$ does not contain all the images of $P = 20\%$). In step 1 the network is fine-tuned with the fine-tune set, resulting in a temporary model (T). T is then trained in step 2 resulting in the final model (F) which is tested on our test set. In step 1, T is trained on 30 epochs with the Stochastic Gradient Descent (SGD) optimizer, a learning rate of 0.0001 and a momentum of 0.9. In step 2, F is trained on 30 epochs with the same optimizer and loss function as in sec.IV-B. When $P = 100\%$ (which means 100% of the images are in the fine-tune set), only step 1 is performed and yields the final model F. When P=0%, the model is the same as in section IV-B. Fig. 9 shows the F1-scores obtained by varying the number of blocks n fine-tuned and P. The simulations are run three times to avoid being blocked in a local minima. Results of VGG16 are not reported for space purposes but they are similar to VGG19.

All the represented models perform better than their baseline when fine-tuning with 100% of the fine-tune set with an exception for MobileNet and DenseNet169 which have a lower F1-score when fine-tuning only 1 block. Increasing the number of blocks to fine-tune improves the performance of MobileNet and ResNet50, especially with a fine-tune set of 100%. We suppose that with the 80% split, we have too many images for the fine-tuning step but not enough for the training step which results in a performance always worse than the baseline.

V. Conclusion

Deep learning approaches appear to outperform human classification in the task of transport mode recognition based on body-worn cameras by 30.09% (Mturk 52.01%, VGG19 82.1%). We speculate one reason for this difference is the amount of example images we presented to the Mturk Workers. In total we provided only 24 example images, in contrast to the 10240 images in the training set used for the computer algorithms. While presenting more images to Workers could improve performance, it poses challenges: users may be disincentivised to spend time looking at additional exemplary images without increased financial compensation. Another explanation for the performance difference is that many pictures have unconventional angles (e.g. tilted cameras) for which it takes conscious effort to understand the visual scene, which is not an issue for a computer algorithm.

Although other work with motion sensors [17] achieved better F1-scores, MTurk is the only vision-based comparison we could retrieve. Also, even though capturing images with a camera is more energy consuming than retrieving sensors modalities, it provides additional contextual information that could help develop more advanced context-aware apps. In a short-term development, we plan to combine the different modalities (sensors, images and audio) available within SHL dataset to reach even higher accuracies.

Without fine-tuning, the best model is VGG16 with a F1-score of 0.791. With fine-tuning, we were able to improve the F1-score to 0.821 with VGG19 using 100% of the fine-tune set and 5 blocks fine-tuned (Fig. 6b). All models still perform better when fine-tuning them. MobileNet and ResNet50 seem to learn better with more fine-tuned blocks.

In a future mobile implementation, we recommend to use MobileNet for space optimization with fine-tuning of 5 blocks using 100% of data. Our findings indicate that computer vision is particularly good at distinguishing Car, Bus, Train and Subway (Fig. 6), for which motion sensors alone had difficulties [11]. This indicates that future work could explore a multimodal fusion approach to further improve performance.

References

[1] J. Engelbrecht, M. J. Booysen, G. van Rooyen, and F. J. Bruwer, "Survey of smartphone-based sensing in vehicles for intelligent transportation system applications," *IET Intelligent Transport Systems*, vol. 9, no. 10, pp. 924–935, 2015.

[2] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Driver behavior profiling using smartphones: A low-cost platform for driver monitoring," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 91–102, Spring 2015.

[3] Y. Vaizman, K. Ellis, and G. Lanckriet, "Recognizing detailed human context in the wild from smartphones and smartwatches," *IEEE Pervasive Computing*, vol. 16, no. 4, pp. 62–74, October 2017.

[4] P. Siirtola and J. Röning, "Recognizing human activities user-independently on smartphones based on accelerometer data," *Int. J. of Interactive Multimedia and Artificial Intelligence*, vol. 1, pp. 38–45, 2012.

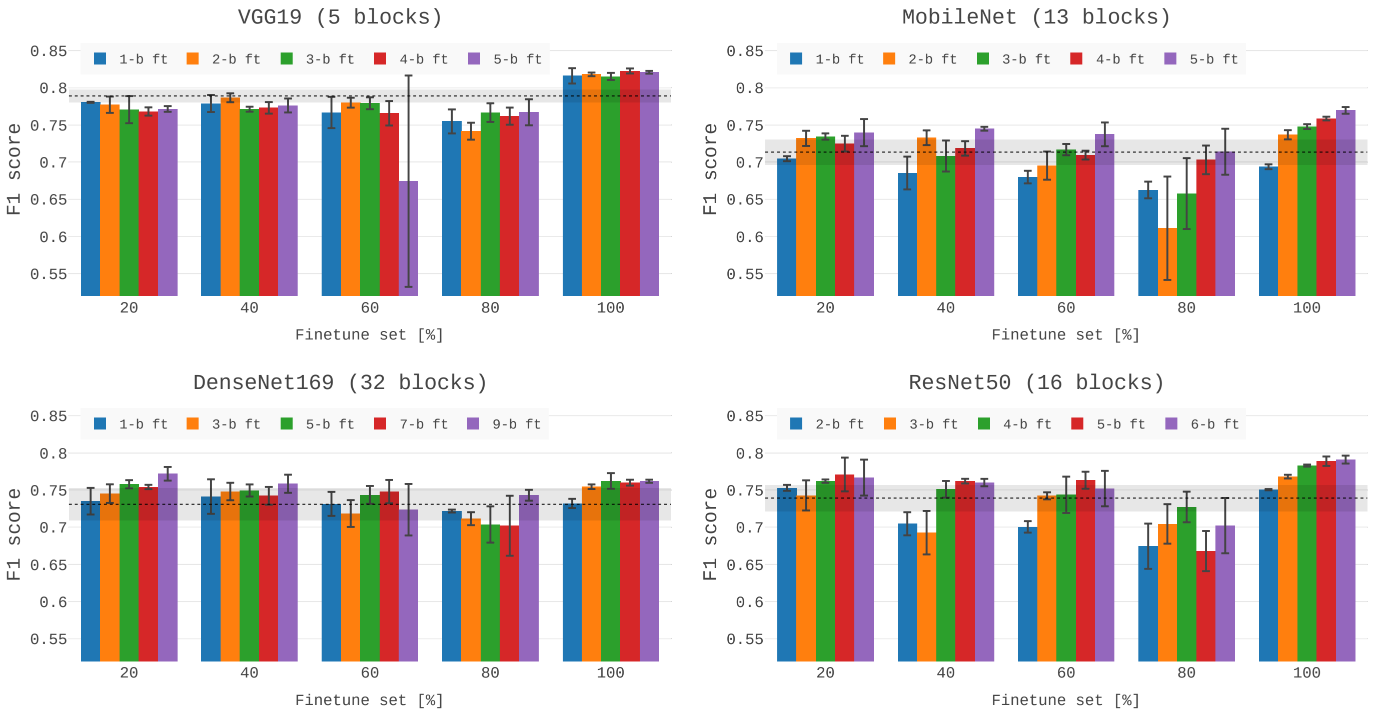


Fig. 9: Fine-tuning of each model. F1-scores are computed on the test set. A percentage of $P\%$ for the fine-tune set means that $100 - P\%$ is used as the train set. The dashed line shows the baseline performance including only step 2 (no fine-tuning, $P = 0\%$). The grey area indicates the standard deviation of this baseline. The number of blocks fine-tuned varies between the models as the total amount of blocks is not the same among each model.

[5] T. Feng and H. J. Timmermans, “Transportation mode recognition using gps and accelerometer data,” *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 118 – 130, 2013.

[6] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in *Proc SenSys*, 2013.

[7] H. Xia, Y. Qiao, J. Jian, and Y. Chang, “Using smart phone sensors to detect transportation modes,” *Sensors*, 2014.

[8] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, “Big data small footprint: The design of a low-power classifier for detecting transportation modes,” *Proc. VLDB Endow.*, vol. 7, no. 13, pp. 1429–1440, Aug. 2014.

[9] X. Su, H. Caceres, H. Tong, and Q. He, “Online travel mode identification using smartphones with battery saving considerations,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2921–2934, Oct 2016.

[10] S. Fang, Y. Fei, Z. Xu, and Y. Tsao, “Learning transportation modes from smartphone sensors based on deep neural network,” *IEEE Sensors Journal*, vol. 17, no. 18, pp. 6111–6118, Sept 2017.

[11] L. Wang, H. Gjoreskia, K. Murao, T. Okita, and D. Roggen, “Summary of the sussex-huawei locomotion-transportation recognition challenge,” in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp ’18. New York, NY, USA: ACM, 2018, pp. 1521–1530.

[12] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen, “Benchmarking the shl recognition challenge with classical and deep-learning pipelines,” in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, ser. UbiComp ’18. New York, NY, USA: ACM, 2018, pp. 1626–1635.

[13] P. Wang and A. F. Smeaton, “Using visual lifelogs to automatically characterize everyday activities,” *Inf. Sci.*, vol. 230, pp. 147–161, 5 2013.

[14] R. Hoyle, R. Templeman, S. Armes, D. Anthony, D. Crandall, and A. Kapadia, “Privacy behaviors of lifeloggers using wearable cameras,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp ’14. New York, NY, USA: ACM, 2014, pp. 571–582.

[15] H. Gjoreski, M. Ciliberto, L. Wang, F. J. O. Morales, S. Mekki, S. Valentin, and D. Roggen, “The university of sussex-huawei locomotion and transportation dataset for multimodal analytics with mobile devices,” *IEEE Access*, vol. 6, pp. 42 592–42 604, 2018.

[16] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, “Squeezing deep learning into mobile and embedded devices,” *IEEE Pervasive Computing*, vol. 3, pp. 82–88, 2017.

[17] L. Wang, H. Gjoreski, M. Ciliberto, S. Mekki, S. Valentin, and D. Roggen, “Enabling reproducible research in sensor-based transportation mode recognition with the sussex-huawei dataset,” *IEEE Access*, vol. PP, pp. 1–1, 01 2019.

[18] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing and Management*, vol. 45, no. 4, pp. 427–437, 2009.

[19] B. S. A. Kittur, Ed H. Chi, “Crowdsourcing user studies with mechanical turk,” *CHI ’08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 453–458, 2008.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[21] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.

[24] G. Huang, Z. Liu, and K. Q. Weinberger, “Densely connected convolutional networks,” *CoRR*, vol. abs/1608.06993, 2016.