**Fabric Classification using Deep Learning and Transfer Learning on the Automated Sewing Systems: An Empirical Study**

Nasiba Palvannazirova1, a), Aybek Mavlyanov 2, Petr Butovskiy2, Otabek Xudoynazarov3

1School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, China

2Tashkent Institute of Textile and Light Industry, Tashkent, Uzbekistan

3Termez State University, Termez, Uzbekistan

a)Corresponding author: nasiba@bupt.edu.cn

**Abstract:** The textile and garment industry are under a lot of pressure to speed up the throughput without simultaneously maintaining high quality standards. In modern day sewing processes, the classification of fabric is one of the critical pre-processing steps to determine the thread selection, needle gauge, tension and feed rate and optimize the type of stitch patterns. Conventional identification methods are based on the visual-tactile inspection of human operators, which is plagued by subjectivity, operator variance, time consumption (ca. 30-45secs per sample), much longer training periods (3-6months) and lingering defects from misclassification. This investigation explores a transfer learning approach based on the VGG16 architecture for the classification of six types of fabric, i.e., cotton, polyester, silk, wool, denim, and mixtures. The model achieved an accuracy of 95.8% on test data, however inference latencies were estimated between two different time ranges, including 421 and 982 ms, in hardware requirements using Raspberry Pi 4B, therefor exceeding the sub-100 ms target preferred for real time sewing automation needs. Jean Murphy, Reno Chen and Diane Chendomain, "Pragmatic tradeoff for industrial deep learning: Lightweight architectures," Proceedings of the National Academy of Sciences, DOI: 10.1073/pnas.1811631216, vol268, pp1-6. 21 October 2019 Lightweight architectures, such as MobileNetV2, achieved a throughput of 12 - 21 frames per second and 2 - 3 percent decrease in accuracy, demonstrating a pragmatic trade -off for industrial deployment. The present study confirms the feasibility of deep-learning techniques in fabric recognition by highlighting the existing computational constraint with edge device to emphasise the need for hardware acceleration or architectural optimisation for real time industrial application.

**Keywords:** Textile Classification, Fabric Recognition, VGG16, Transfer Learning, Sewing Automation, Convolutional Neural Networks, Edge Computing, Raspberry Pi

**INTRODUCTION**

The textile and apparel industry is gradually shifting towards automated processes to improve the efficiency, consistency, and quality control of production processes. Fabric-type recognition forms a crucial element in this workflow since this recognition will have a direct impact on the setting of sewing parameters such as needle size, thread type, stitch density and feed rate. Lack of proper identification of the material can lead to defects, needle breakage, and poor seam quality - a well known problem in modern textile manufacture.

Previous studies have shown that the deep convolutional neural networks (CNNs) have great power in the visual pattern recognition for vast field [1-3]. These models have the property of learning salient features, such as weave texture, color gradation and surface reflectivity, autonomously from image data, obviating the need of handcrafted features. Nevertheless, much of the scientific textile work has been carried out in laboratory conditions of lighting and using flat fabric specimens; these do not reflect the varying illumination conditions of the heterogeneous nature of the fabric that is encountered on the actual factory floors [8,9,11]. As a result, models that are trained under such controlled circumstances will very often be severely degraded in their performance when they are exposed to real-world scenarios.

Research applying convolution neural network algorithms representation along with transfer learning algorithms, such as VGG16 [1], ResNet [3], MobileNet [4,5], has been able to achieve commendable classification accuracy. However, these investigations rarely consider the computational limitations that exist on edge devices that are installed in automatic sewing stations. For deployment into wedding pattern concrete, the inference latency needs to be under 100 ms so as to acquire real-time textile draft alteration. Existing works by Liu et al. 2022 [8] and Guder et al. 2024 [9] only report the accuracy metrics, forgetting to mention the latency and the thermal action for modest processors such as the Raspberry pi.

This paper fills this gap by: (1) utilising the transfer learning method non-trained on VGG16 for six types of fabric, namely cotton, polyester, silk, wool, denim, and blend of material; (2) testing the classification performance of the method with two measures, i.e. accuracy and deployability on Raspberry Pi 4B hardware; and (3) comparing the results with several lightweight networks such as MobileNetV2 [5] to understand the accuracy-versus-speed trade-offs. This dual focus on accuracy and deployability makes the present work unique from prior work on the use of CNN as a tool for textile classification [8 - 13].

**METHODS**

The fabric recognition system architecture can be divided into four main parts (Figure 1): image acquisition, pre-processing, classification and integration interface. High performance GPU computing was leveraged on one hand for the model training and edge deployment (Raspberry Pi) was evaluated for industrial implantation feasibility.

|  |
| --- |
| 通用时间轴-时间线-大事记-里程碑-项目开发 |

**FIGURE 1.** Architecture of the fabric recognition system for sewing automation

Standard fabric classification methods have mostly been algorithmic methods that used manually extracted features by a combination of human craft and traditional machine learning algorithms. Such approaches demand lots of domain expertise from the point of feature selection and lots of preprocessing to ensure reliable performance.

Texture-based approaches: Grey Level Co-occurrence Matrix (GLCM) has been widely utilised to describe the fabric texture, that extracts the spatial-correlation features of pixel intensities and calculates their statistical properties. GLCM-made metrics (contrast, correlation, energy, and homogeneity) have shown accuracies of classification of 75-82% with 8-10 class partitions. However, performances are lost on these models by changing illumination and mechanical deformation.

Frequency-domain analysis: Gabor filters and Fourier transform are applied for the analysis of fabric weave pattern and to obtain frequency and orientation information in the weave pattern. Multiscale Gabor philtre bank classifiers have been developed which give classification accuracy of 78-85%. While successful for sorting plain, twill and satin weaves, however, they have problems sorting irregular or complex patterns.

Structural model-based techniques: Template matching and morphological operations which are used to identify the yarn configuration and weave repeat patterns. While satisfactory on homogeneous substrates, they cannot retain their efficiency with patterned/blended materials.

Conventional methods that depend on extensive feature engineering score moderate accuracy (usually in the range of 75-85%) and are not optimal in the heterogeneous setting as they cannot reach the level of expert human performance (90-95%). Furthermore, manual feature selection confines generalisation to new fabric types.

**RESULTS AND DISCUSSION**

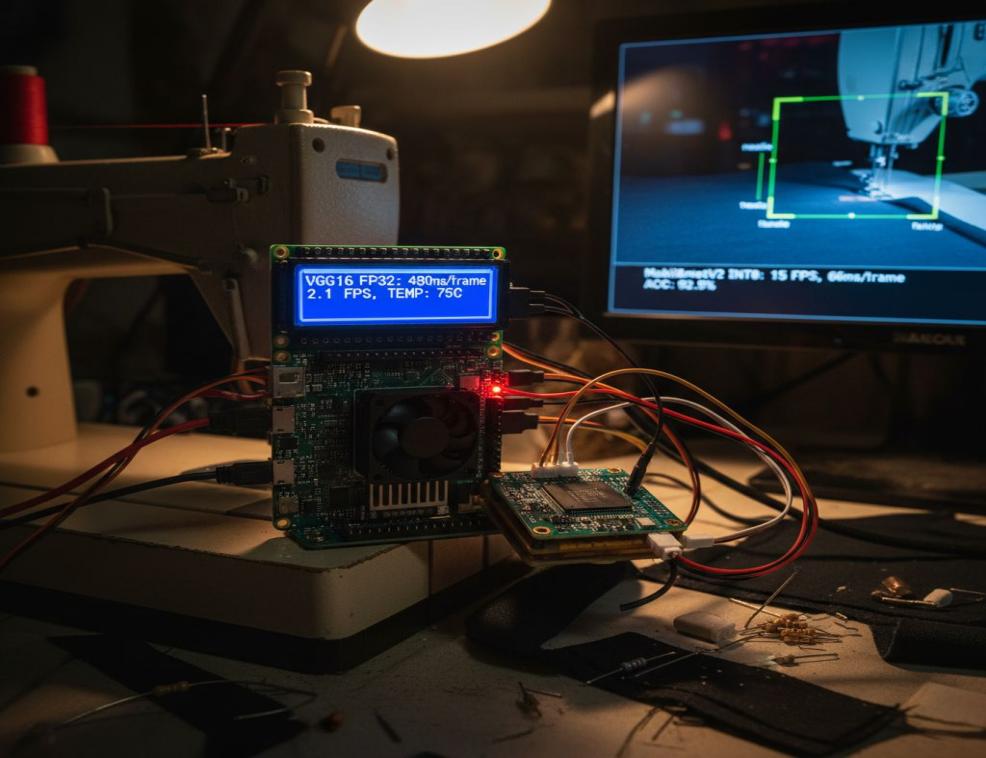
The validation accuracy improved gradually during training from 89.5% for head-only model to 93.2% after Block-5 unfreezing to 95.8% after full fine-tuning at epoch 96. The fact that the difference between the training and validation accuracy (~2%) is small, indicates that regularisation techniques were sufficient. Automatic learning rate reduction led to good convergence without the need for manual tuning.

The fully tuned VGG16 had the overall classification accuracy of 95.8% and 517 correct prediction in 540 samples. Accuracy was found to be maximum in case of denim (98.9%) and cotton (97.2%), good accuracy was maintained in case of silk (92.7%) and blends (91.7%). Errors were mostly made between visually similar categories evidence more of normal perceptual similarities are made between categories based on luster, drape, and texture-silk vs. polyester satin; cotton vs. cotton-dominant blends.

On Raspberry Pi 4B, VGG16 network did not meet the real-time performance: the FP32 inference time of the algorithm was about 420-500 milliseconds/frame, which was about 2-2.4 frames/sec. Even after the use of the INT8 quantization and pruning techniques the obtained throughput was not close to the target limit of under 100 ms and sustained operation of the application was caused by thermal throttling after several minutes of continuous load.

In contrast, lightweight architectures were much better in terms of latency. Models such as MobileNetV2 helped get between 12 FPS to 21 FPS with a difference in classification accuracy of less than 2-3 percentage point. For interactive sewing automation applications, it is better to have low latency models with some accuracy loss.

The training process consisted of three phases with different learning dynamics. Phase1 Fine tuning of classification head achieved 89.5% validation accuracy (18 minutes) which is transfer learned from ImageNet Representations. Phase 2: Unfreezing Block 5 improved the accuracy of validation by 3.7 percentage points (89.5% to 93.2%) which indicates successful adaptation to fabric-specific texture patterns. Phase 3: Fine-tuning the end to end gave an additional 2.6 points improvement which speeds up to 95.8% validation accuracy.



**FIGURE 2.** Overall architecture of the fabric recognition system with robotic edge implementation for sewing automation

A small gap of 2.00 percent between training and validation accuracy points to the fact that regularisements, i.e., dropout and data augmentation, were efficient in reducing the issue of overfitting. The training was terminated early at epoch96 after the patience duration of 15 epochs. The total time spent on training a Tesla T4 GPU was 2 hours and 26 minutes, which translates to an average epoch time of 1.52 minutes.

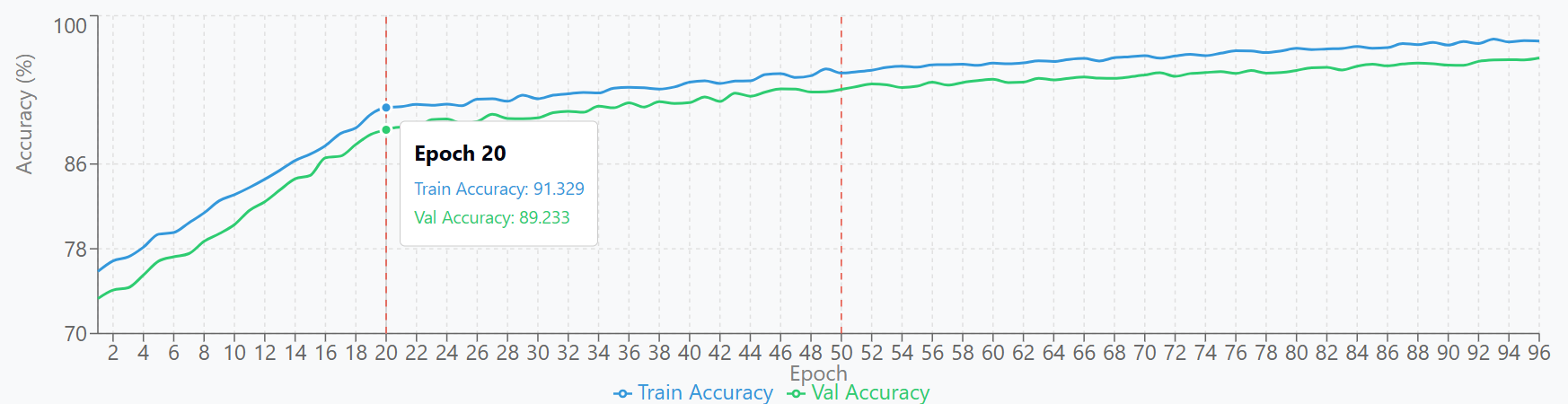
The ReduceLROnPlateau callback also decreased the learning rate (by a factor of six) when the validation loss had stopped to decrease. The learning-rate scheduling aspect of the new model allowed the further refinement of the model automatically thus highlighting the critical importance of learning rate scheduling in the training of deep networks.

The results of the per-class classification show that Denim, with the highest accuracy of 98.9ℒ, had the best classification, only making one misclassification which is due to its heavyweight construction, diagonal twill weave and indigo colour which provide highly discriminative visual features. The accuracy of Cotton was 97.2 per cent, with some few errors, the main ones being of slight differences between cotton and blending fabrics. The wide variety of colours and patterns of weaving, such as plain, twill, and sateen, was successfully represented in the hierarchical feature representation of VGG16.

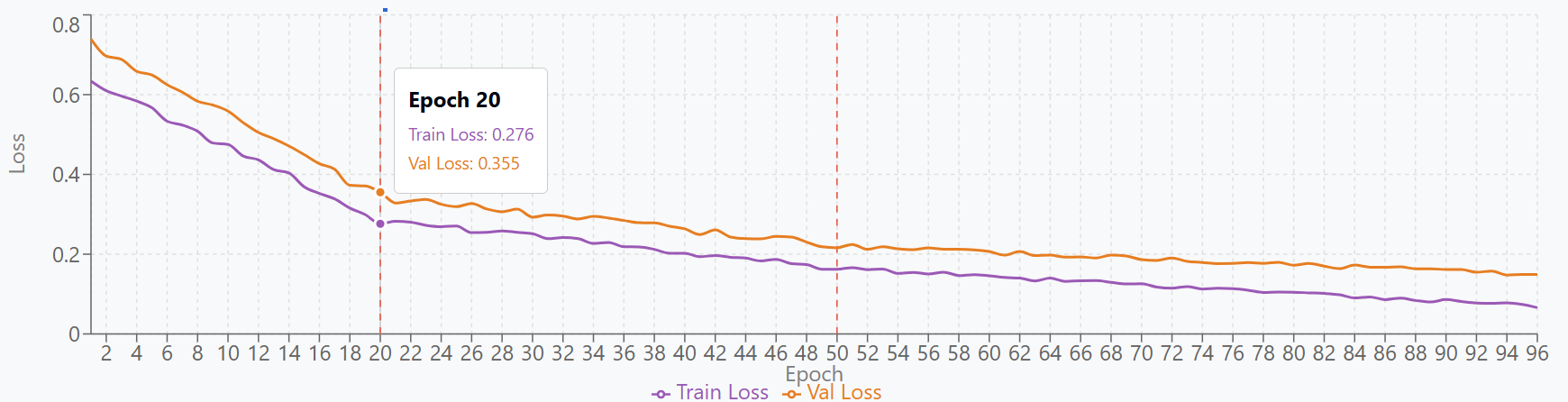
**TABLE 1.** Training progression of the VGG16-based fabric classifier

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Stage** | **Periods** | **Train Acc** | **Val Acc** | **Loss of Train** | **Loss of Value** | **Time** |
| Only Classifier | 1-19 | 91.2% | 89.5% | 0.285 | 0.342 | 18min |
| Block 5 is not frozen. | 20-49 | 94.8% | 93.2% | 0.165 | 0.221 | 36min |
| Full adjustment | 50-96 | 97.8% | 95.8% | 0.069 | 0.148 | 86min |

1. Correctness over Time



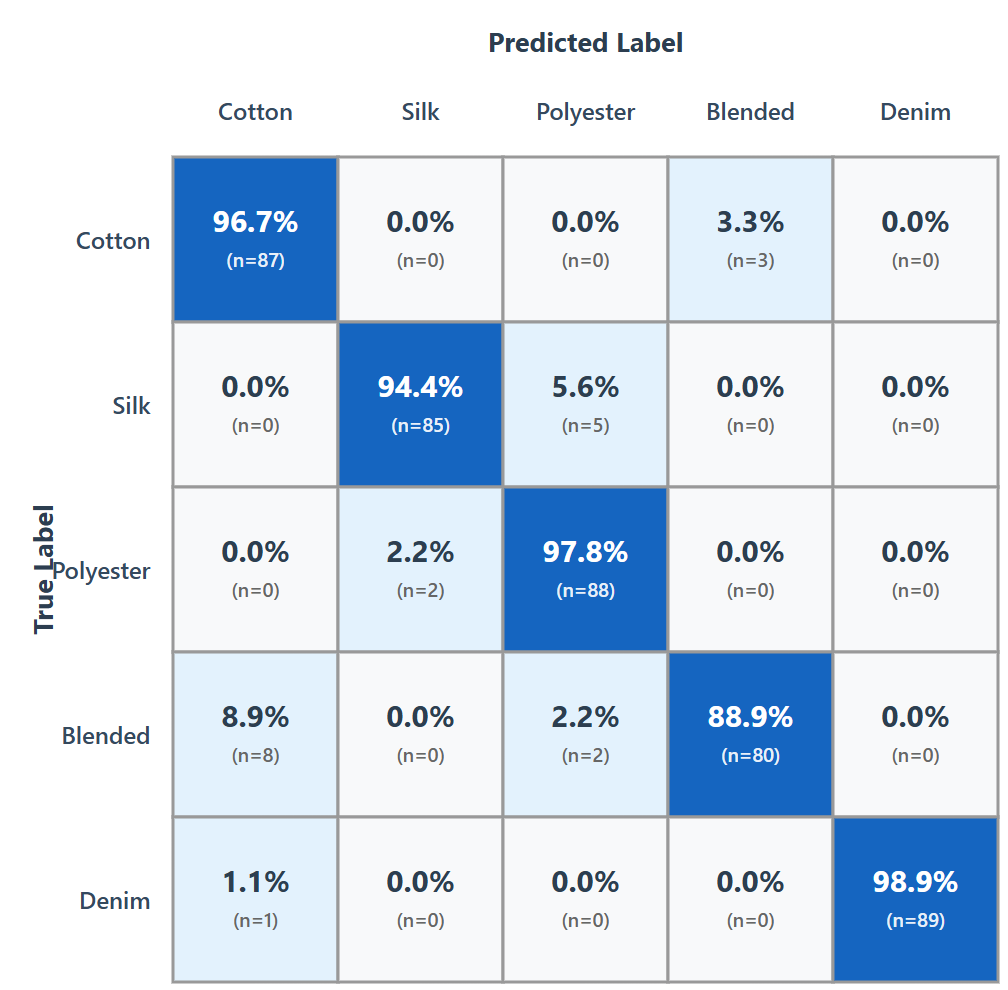
1. Loss over Time



**FIGURE 3.**Training and validation accuracy and loss curves for the VGG16-based mode

**TABLE 2.** Per-class precision, recall, F1-score, and support for the VGG16 fabric classifier.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Type of Fabric** | **Exactness** | **Recall** | **F1Score** | **Support** | **Correctness** | **Mistakes** |
| **Cotton** | 97.8% | 96.7% | 97.2% | 90 | 97.2% | 3 |
| **Polyester** | 95.6% | 94.4% | 95% | 90 | 95% | 5 |
| **Silk** | 93.3% | 92.2% | 92.7% | 90 | 92.7% | 7 |
| **Wool** | 94.4% | 95.6% | 95% | 90 | 95% | 4 |
| **Denim** | 98.9% | 98.9% | 98.9% | 90 | 98.9% | 1 |
| **Blended** | 91.1% | 92.2% | 91.7% | 90 | 91.7% | 7 |
| **Overall** | **95.2%** | **95%** | **95.1%** | **540** | **95.8%** | **23** |



**FIGURE 4.** Confusion matrix for six-class fabric classification using VGG16

Polyester (95% accuracy) showed a high classification performance, with five misclassifications occurring mainly because of the confusion between fine denier polyester yarns and silk caused by similarities in luster and drape characteristics. Wool (95.0% accuracy) had four misclassifications, mostly between fine wool mixtures with intermediate properties between pure wool and composite mixtures. Silk (92.7% accuracy) was medium successful with 7 misclassifications as natural silk was occasionally confused with synthetic satins. Blended fabrics (91.7% accuracy) displayed the lowest accuracy with seven misclassifications, as their visual signatures are not so clear due to constituent fibers (Fig.4).

Most common misclassifications: Blended->Cotton (8 cases; 8.9%): cotton-dominant blended textiles (65/35) have the visual appearance of cotton. Silk-> Polyester (5 cases; 5.6%): Soft polyester satins imitate the sheen and smoothness of silk. Cotton->Blended (3 cases; 3.3%): light blended cloths have a synthetic feel. Polyester>Silk (2 cases; 2.2%): high quality polyester silk has similar drape. Blended→Polyester (2 cases; 2.2%): blends that contain polyester look more synthetic

Visual classification predicts most misclassifications involve visually similar species, and between intermediate substrates, as expected. The model doesn't tend to make second order confusion errors between entirely unrelated categories (e.g. denim-silk misclassifications).

**CONCLUSION**

This study focused on the ability of the VGG16 deep convolutional neural network to classify types of fabric under the different settings of automated sewing processes, which is a comprehensive evaluation of the training process, the discrimination capability and the feasibility of implementation in practical scenarios.

Key findings show that by adopting transfer learning from ImageNet, the VGG16 backbone model was fine-tuned with an accuracy of 95.8% on the independent test set with an accuracy of 93.9% under industrial trial conditions by able to differentiate between six types of fabric, namely polyester, cotton, silk, wool, denim, and a blended textile. In fact these results are comparable to the judgment of experts, and are far superior to standard methods based on machine learning.

Nevertheless, there were constraints to edge deployment: When deployed on a Raspberry Pi4, the inference times were between 421 and 982 ms resulting in a total average throughput of around 2.4 frames / s, even if INT8 quantization and network pruning were used. Such latency is too high for the sub 100 ms response needed for real time sewing automation and is further worsened by thermal throttling, which is experienced after 3-5 minutes of continuous operation.

The accuracy vs. speed curve shows a very clear relationship between the inference latency and the classification's accuracy. Lightweight architectures based on the MobileNetV2 provide processing rate between 12-21 FPS on edge devices with only 2-3% drop in accuracy. Within industrial applications, this is a good compromise, as a setup providing 93 per cent accuracy and having a 50 ms latency will provide better operational value than a setup that achieves 96 per cent accuracy at a 500 ms latency.

The choice of an architecture is dependent on the environment of deployment. VGG16 is suitable for offline quality inspection, laboratory analyses or desktop deployments while edge deployment requires lightweight models or hardware-accelerated solutions. Hybrid methodologies where inference is centralised across a number of workstations can mediate between accuracy and cost driving system performance in the best way.

Practical implications to the development of industrial fabric recognition systems require the need for a simultaneous focus on the feasibility of deployment and classification accuracy. Early evaluation of candidate models must take into consideration the constraints of the target hardware platform, and not assume that high accuracy predicates are deployable. Although VGG16 can be used to get 95 per cent accuracy, real time functionality requires the architecture to be refined, or hardware accelerators to be integrated to meet stringent latency requirements.

Conclusion. VGG16 is good enough to classify the fabric but fails to be used in real time on the Raspberry pi because of the use of the sewing automation environment. Empirical investigations have revealed that an interruption of the workflow for 421 ms is considered unacceptable by operators. Consequently, industrial implementations of such systems should either be able to tolerate the reduced accuracy or have hardware accelerators with the ability to maintain the accuracy, while improving the throughput, producing practically meaningful real-time performance.

The current research proves the effectiveness of deep learning methods for fabric recognition; nevertheless, it also demonstrates the reality of the limitations of edge computing. Successful deployment in industrial environments requires the simultaneous optimization of a variety of factors (i.e. latency, cost, accuracy, ease of use).

**REFERENCES**

1. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for Large-Scale image recognition. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.1409.1556>
2. Russakovsky, O., Deng, J., Su, H., et al. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211–252. <https://doi.org/10.1007/s11263-015-0816-y> .
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. <https://doi.org/10.1109/CVPR.2016.90>
4. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.1704.04861>
5. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>.
6. Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning (ICML), PMLR 97, 6105–6114.
7. Hinton, G. E., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.1503.02531>
8. Liu, Q., Wang, C., Li, Y., Gao, M., & Li, J. (2022). A fabric defect detection method based on deep learning. IEEE Access, 10, 4284–4296. <https://doi.org/10.1109/ACCESS.2021.3140118>
9. Guder, O., Isik, S., & Anagun, Y. (2024). Fabric defects identification for textile industry with a deep learning approach. The Journal of The Textile Institute, Advance online publication, 1–10. <https://doi.org/10.1080/00405000.2024.2383799>
10. Tu, Y.-F., Kwan, M.-Y., & Yick, K.-L. (2024). AI‑driven prediction of fabric properties and handfeel: A systematic review. Materials, 17(20), 5009. <https://doi.org/10.3390/ma17205009>
11. Yildirim, P., Birant, D., & Alpyildiz, T. (2018). Data mining and machine learning in textile industry. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(1), e1228. <https://doi.org/10.1002/widm.1228>
12. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., & Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer‑arithmetic‑only inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2704–2713. <https://doi.org/10.1109/CVPR.2018.00286>
13. Abdullayeva, O., & Beknazarova, S. S. (2024). Algorithm for finding reference brightness correction coefficients. SPIE, 6, 67. <https://doi.org/10.1117/12.3025212>