Comparison of Training for Hand Gesture Recognition for Synthetic and Real Datasets

Pranav Vaidik Dhulipala, Samuel Oncken, Steven Claypool, and Stavros Kalafatis

Department of Electrical and Computer Engineering, Texas A&M University, College Station, Texas-77845, USA

Abstract. Human gesture recognition is often implemented in many HRI applications. Building datasets that involve human subjects, when aiming to capture comprehensive diversity and all possible edge cases is often both challenging and labor-intensive. While applying the concept of domain randomization to build synthetic datasets helps address the problem, an innate reality gap always exists that needs to be mitigated. In this paper, We present and discuss a comprehensive performance comparison of our synth datasets with real ones and demonstrate the results in this paper.

1 Introduction

In recent years, the application of deep learning and machine learning techniques has become paramount for various computer vision applications such as object recognition, localization, and segmentation [34]. A considerable time is often spent by many deep learning practitioners in finding suitable datasets for their projects, followed by the necessary pre-processing and data-cleaning processes before applying any of the machine learning techniques [12, 19].

Furthermore, for human-centric computer vision applications such as hand gesture recognition, pose detection, and localization, an often challenging step of the process is obtaining suitable datasets that satisfy the annotation requirements, mainly due to the human body's fluidity in shape, having many degrees of freedom with various of joints, unlike most solid objects [28, 27]. Moreover, annotations can drastically vary with activity and gesture recognition applications as per project requirements, with different interpretations conveyed by the same pose or gesture, especially in the case of sign language. As a result, many machine learning and deep learning practitioners often build and annotate custom datasets for human-centric computer vision applications according to their requirements.

Common practices for building custom datasets often involve a tedious process of capturing pictures or videos of poses from either a set of limited subjects under controlled conditions or the practitioners themselves, usually indoors [32, 30, 43]. As a result, the custom datasets often lack scale and variability, rendering them impractical for reuse in other projects depending on complexity, requiring the practitioners to repeat the steps for newer projects.

While practices such as web scraping from an image search engine or acquiring video snippets from video-sharing platforms such as YouTube can solve the problem of scale, the cleaning and annotation process at such a high scale is extremely time-consuming [6], with practitioners often having tens of thousands of images or video frames to process. Real et al. [33] presents various stages and numerous challenges in cleaning, selecting, and annotation of data from YouTube videos for computer vision applications. Depending on the practitioner, human errors in annotations are also possible, which can cause another set of problems with the model performance [31, 35].

The recently emerged domain randomization technique addresses the above-mentioned challenge of annotation and scale by proposing to train the deep learning models with

David C. Wyld et al. (Eds): SIGML, CNSA, MoWiN, CCSEIT, NIAI, AIAP – 2025
pp. 179 -193, 2025. CS & IT - CSCP 2025 DOI: 10.5121/csit.2024.150214

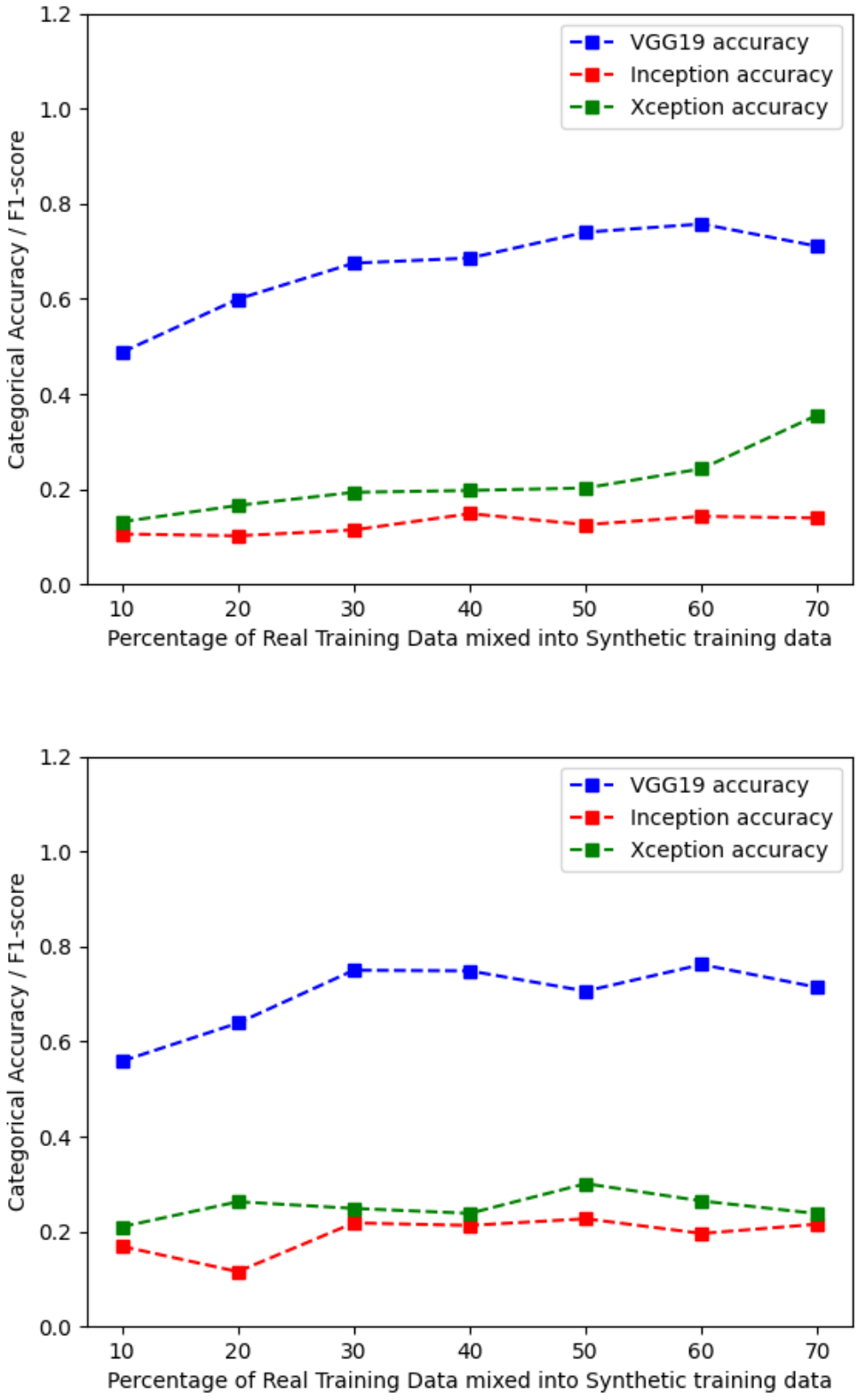

Fig. 1. Comparison of model performance trained for accuracy metric with the synthetic train set with varying mix percentages of real train data with convolution weights trained (top) from scratch with data augmentation (bottom)from scratch without data augmentation

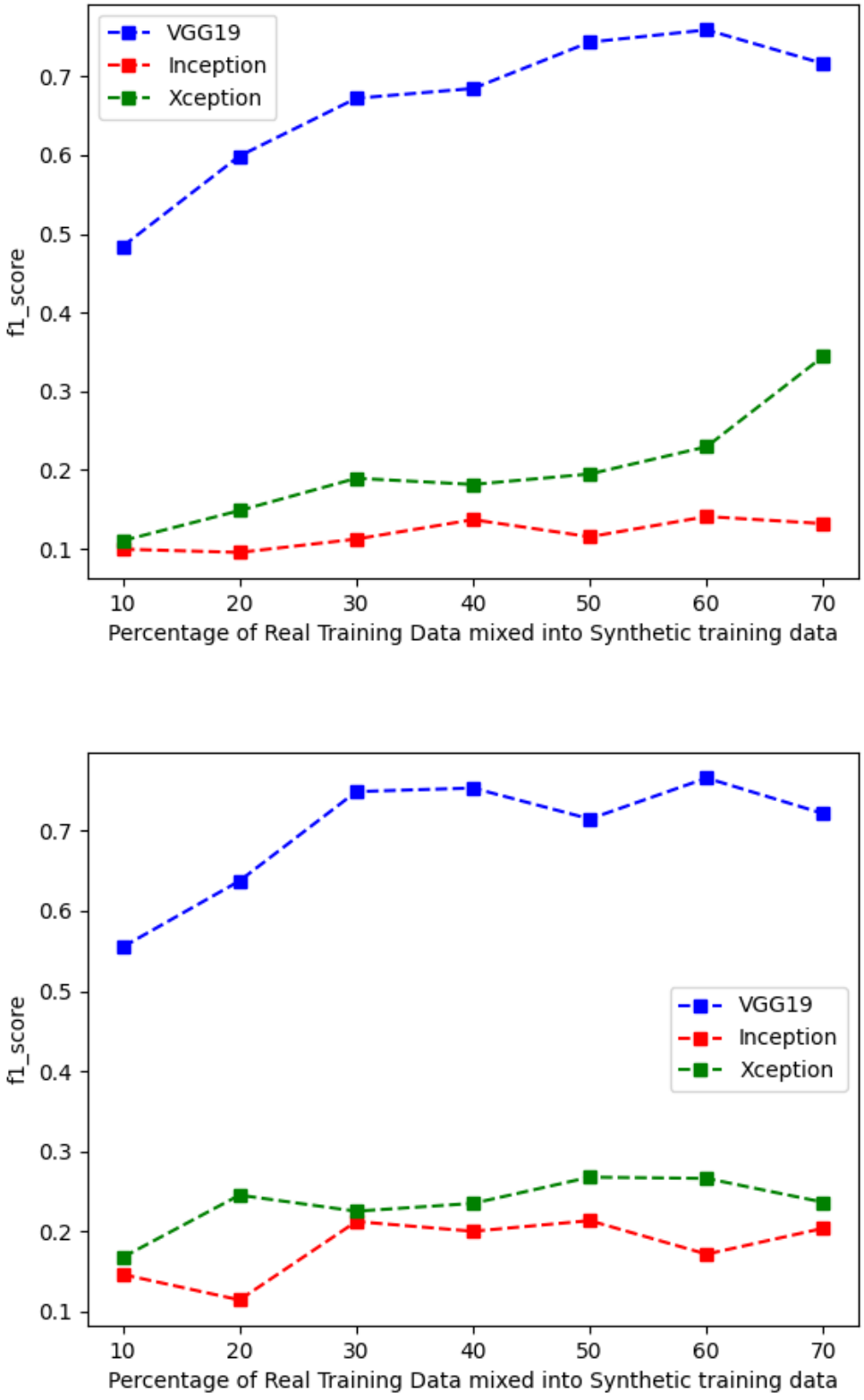

Fig. 2. Comparison of model performance for F1 score metric trained with the synthetic train set with varying mix percentages of real train data with convolution weights trained (top) from scratch with data augmentation (bottom)from scratch without data augmentation

simulated data that transfer to real data [38]. The technique emphasizes randomizing rendering during training while adding variability in the simulation to the extent that the real world may also appear as just another variation to the simulation. However, there is also a reality gap [38] problem that accompanies the use of synthetic datasets for training, when simulators are unable to produce the photo-realistic textures and lighting that match their real counterparts. Several publications in the recent literature have addressed the reality gap problem [39, 41, 13, 15, 25], proposing methods to mitigate this problem.

Recent literature has seen an emergence of several synthetic datasets for various computer vision applications such as object recognition [1, 20, 14, 16], gait detection [9], and semantic segmentation [2, 3]. Pipelines to build synthetic data such as Unity Perception toolkit [40] were also proposed for many applications.

In our recent work, we have introduced a pipeline for building synthetic image datasets for hand gesture recognition [7], while introducing three synthetic datasets built using the pipeline, similar to the Sign Language for Numbers (Digits Dataset) [18], American Sign Language dataset [22], and HANDS dataset [26] respectively, and discussed the simulation environment, generation parameters, and synthetic dataset comparison with their real counterparts. The datasets are publicly available on IEEE Dataport [8].

In this work, we present the comparison of the digits dataset from [7] with its real dataset counterpart Sign Language for Numbers dataset [18], and discuss the performance comparisons for gesture recognition.

2 Related Work

The foundational principle of domain randomization[17] has been exploited to study and publish several human-centric synthetic datasets and respective pipelines over recent years.

The SURREAL dataset introduced by Varol et al. [42], presented a dataset containing more than 6 million frames of synthetic human activity. Their work resulted from the application of the motion capture data from the CMU motion capture dataset (http://mocap.cs.cmu.edu/) onto several SMPL-generated human body models [23] and annotated for semantic segmentation of body parts and human pose estimation. Nevertheless, while off-the-shelf motion capture datasets expedite the data generation process, custom projects may have difficulty obtaining a dataset with the required custom hand gestures.

The Sans People generator[10] was introduced by Unity Technologies for generating various human-centric datasets for computer vision detection, localization, segmentation, and pose estimation. The generator consists of a unity scene with a virtual background screen with rapidly changing background wallpapers, and spawning human models with random poses and obstructive objects with random orientations. Annotated data is generated and captured during the scene for various computer vision applications. However, the generator has limitations that make it challenging to confine the human models to a predefined set of poses.

A synthetic sign language dataset generation pipeline for gesture recognition was discussed in Miura et al. [24], where they applied a set of sign language poses onto a set of synthetic human models from the SURREAL dataset. The dataset, however, consists of only full-body images, while a common practice with hand gesture datasets is to capture only hand signs. Moreover, the dataset does not use varying background textures to add variations to the dataset.

A synthetic hand gesture generating tool was introduced by [11], but the dataset generated is limited to driving scenarios. Furthermore, the tool generates images that mimic depth and infrared cameras.

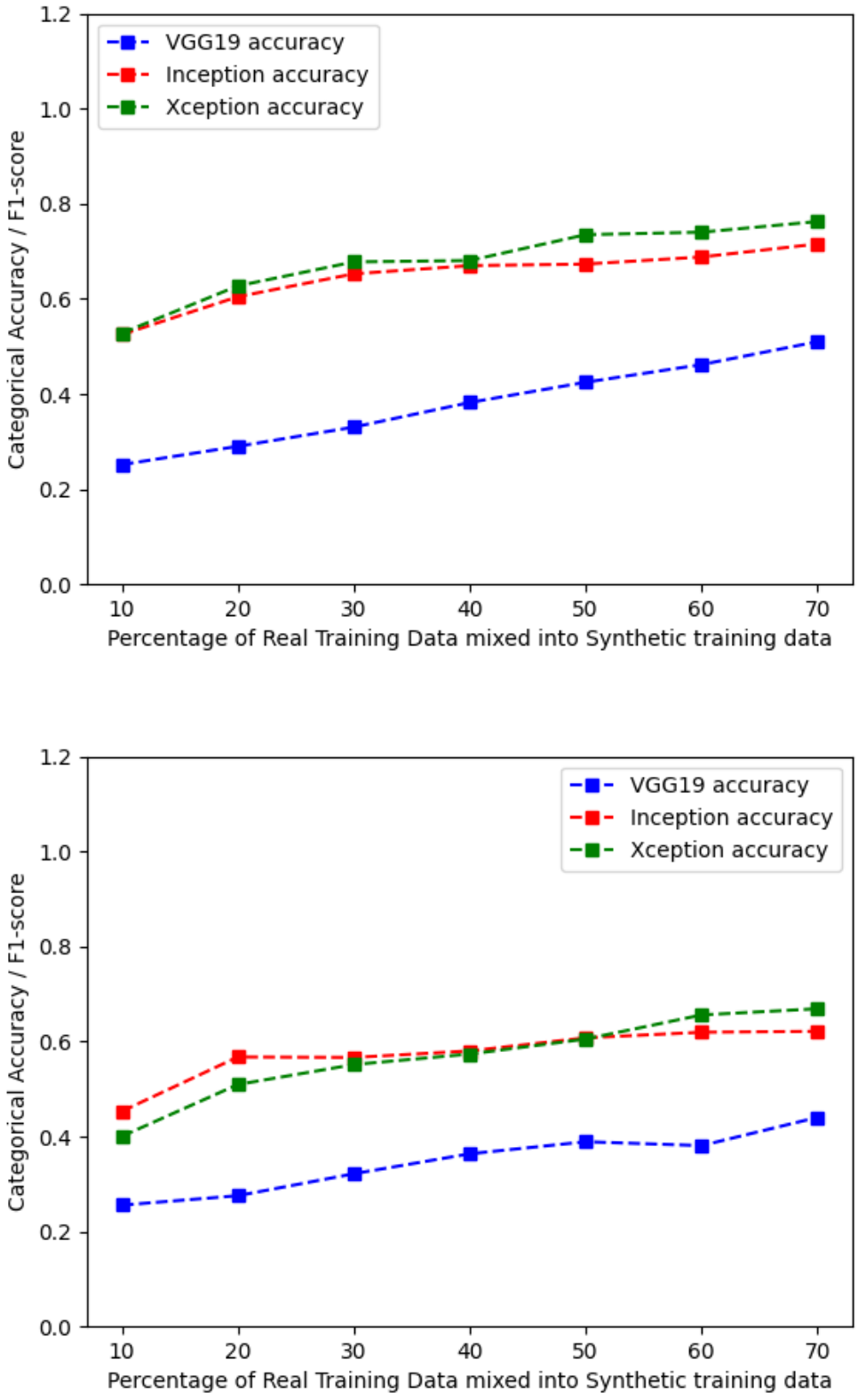

Fig. 3. Comparison of model performance trained for accuracy metric with the synthetic train set with varying mix percentages of real train data with convolution weights trained (top) with transfer learning and data augmentation (bottom) with transfer learning without data augmentation

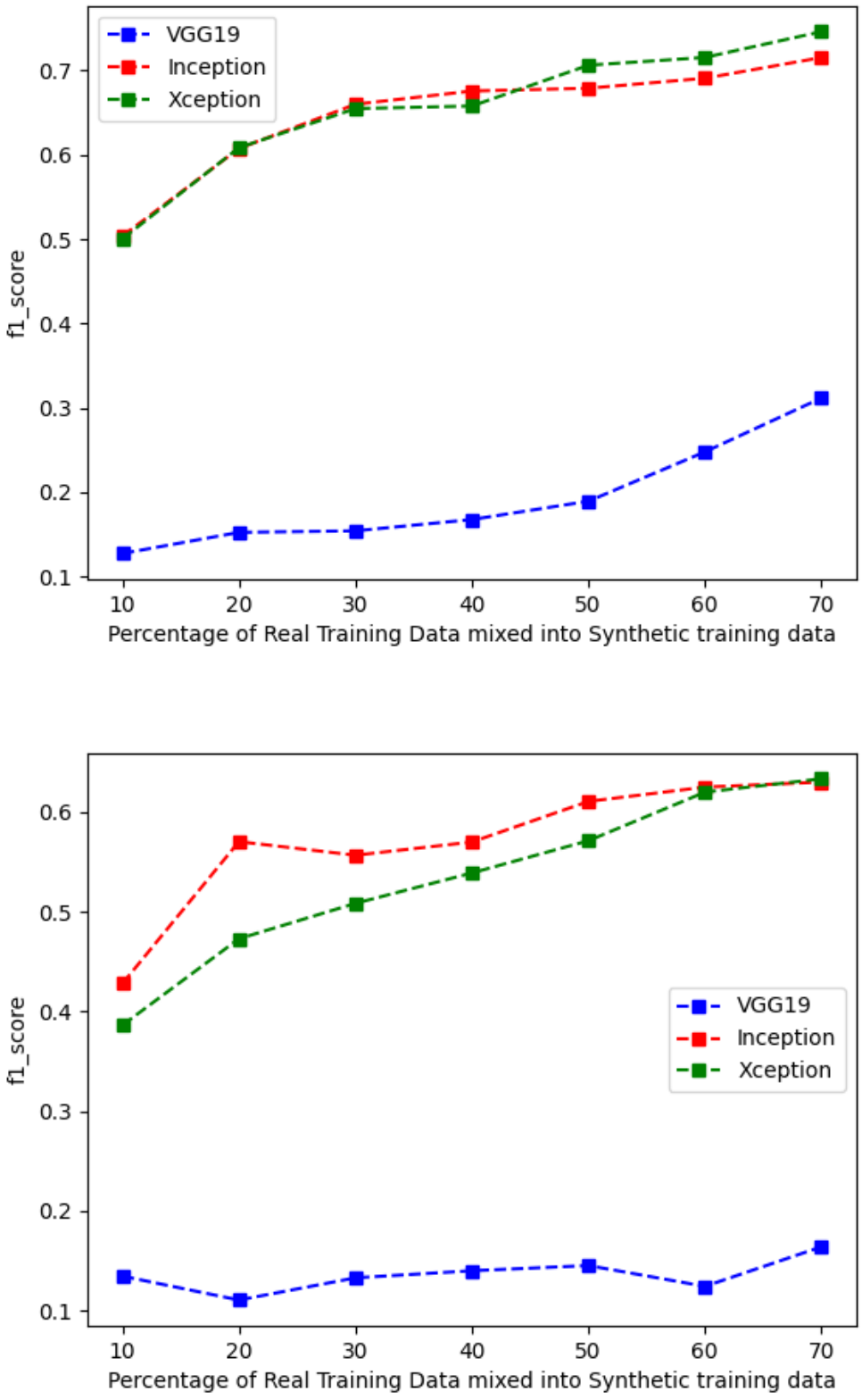

Fig. 4. Comparison of model performance trained for F1 score metric with the synthetic train set with varying mix percentages of real train data with convolution weights trained (top) with transfer learning and data augmentation (bottom) with transfer learning without data augmentation

Model	Real		Synthetic		Mix 10%		Mix 70%	
	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
VGG19	0.75	0.76	0.12	0.12	0.49	0.48	0.74	0.72
Inception-v3	0.09	0.0	0.09	0.04	0.11	0.10	0.14	0.13
Xception	0.09	0.0	0.11	0.10	0.14	0.11	0.36	0.34

Table 1. Model performance comparison with Data Augmentation and model weights trained from scratch

Lindgren et al.[21] also discussed a framework to learn hand gestures using synthetically generated training samples, however, the framework only used one humanoid robot model with depth images for the classification problem.

3 Methods

As mentioned in the introduction, we use the digits dataset from [7] for the classification problem. Training methods, optimization parameters, performance metrics, and methods are further discussed here.

We have used VGG19 [36], Inception-v3 [37], and Xception [4] model architectures, some of the commonly used models for image classification, to perform on both the real and synthetic datasets for comparison. For each model, the original dense layers of the architecture are replaced with new dense layers to fit the number of classes in the digits dataset.

To compare the model performances on the datasets, we have decided to use the performance against the test set of the real dataset for all the models. The rationale for this is that the models are trained to work with real-world data, and also help understand the effect of synthetic data. Furthermore, this method of evaluation on mixed datasets provides more insight into their effect in addressing the problem of the reality gap for classification problems. Also, testing all models against one test set helps us evaluate all the methods fairly.

Each model is set to be trained with and without transfer learning [29]. In the case of the former, Imagenet weights [5] have been used on the convolutional layers, and the weights in the convolution layers are set to be frozen for this case. The models are also trained with and without the presence of data augmentation. Width shift, height shift, rotation, and sheer variations to the augmentation. However, we did not use vertical and horizontal flips to the variations, since all images are left-handed in the datasets. Considering the augmentation and transfer learning variations, we train 4 models per model architecture, totaling 12 trained models for each dataset. All models use the Adam optimizer in this study. The learning rate for the optimizer is set to 10^{-5} and categorical cross-entropy is used for the loss.

All datasets are set to be split into 60% train, 20% validation, and 20% test sets respectively. To add more variations to the study, mix ratio datasets were built by mixing the synthetic dataset with various mix ratios of the real dataset to study their model performances along with that of the real and synthetic datasets. In this study, the mix ratios are varied from 10%-70% of the real dataset, with increments of 10%, totaling 7 mix ratio train sets for evaluating the effect of the mixing.

With the addition of the mix train sets, to the real and synthetic datasets, we have trained a total of 9 training sets, each trained with 12 different models, totaling 108 models for performance comparison. Various performance trends for the same are shown in the discussion section.

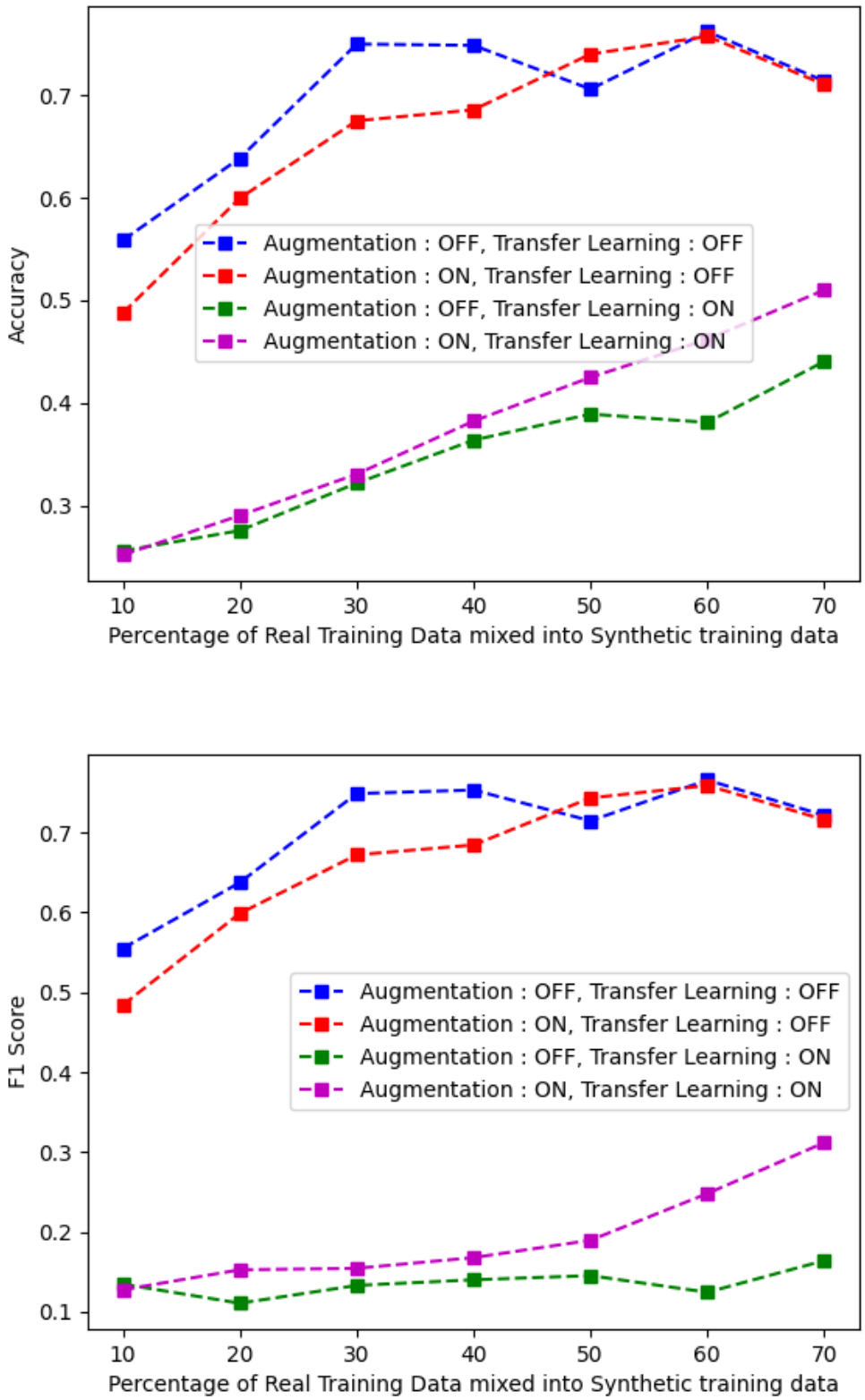

Fig. 5. comparison of VGG19 model performance in different cases of data augmentation and transfer learning (top) Accuracy (bottom) F1 score

Since the models are trained for classification, the performances are evaluated using categorical accuracy and F1 score, both commonly used metrics for robust classification problems. Performance metrics are shown in tables 2-4, discussed in the next section.

4 Discussion

Model performances on the datasets were compared with various parameters, as explained in this section. We focus on the effect of transfer learning, data augmentation, and the effect of mixing various percentages of real training data into the synthetic dataset on the model performances. Performances are measured using the accuracy and F1-score metrics for all cases.

Model	Real		Synthetic		Mix 10%		Mix 70%	
	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
VGG19	0.62	0.13	0.23	0.20	0.25	0.13	0.51	0.31
Inception-v3	0.80	0.80	0.25	0.24	0.53	0.50	0.72	0.72
Xception	0.85	0.84	0.29	0.25	0.53	0.50	0.76	0.75

Table 2. Model performance comparison with Transfer Learning and Data Augmentation

Figures 1 and 2 shows the performance results of the models with accuracy and F1-score metrics respectively. As seen in every case, the performance generally improves with an increasing mix ratio of the real train dataset. It can also be noted that performances in both metrics show very similar trends. The difference between the lowest and highest ratios of mixing in all cases was as high as 0.3 for both accuracy and F1-score metrics. This shows the effect of the reality gap, while also showing that it can be mitigated with the mixing of the real data into the synthetic train datasets while training the models, to improve the model performance.

As mentioned in the methods section, all three models were trained with and without transfer learning. For consistency, only the weights in the convolution layers of the models were frozen in the case of transfer learning for every model, while the dense layers were unfrozen in all cases. Imagenet weights are used for all transfer learning cases.

From figures 1 and 2, it can be observed that VGG19 model architectures show significantly better performance compared to that of Inception-v3 and Xception architectures when their weights are trained from scratch. Both accuracy and F1 score differences were as high as 0.3 for every mix ratio.

Data augmentation's effect on performance shows improvement in performance in all, if not most cases. This is true for every case of the mix ratio in the train set, as seen from comparing the plots in figures 1, 2, 3, and 4 respectively. The difference in performance was as high as 0.1 for both metrics with the addition of data augmentation to the training process. Hence we can recommend data augmentation to be used while training for better results. Another interesting observation was that data augmentation's effect on improving performance was considerably higher than increasing the mix ratio, especially in cases with low model performance. This is true irrespective of transfer learning employed.

Furthermore, plots from figures 5, 6, and 7 show the model performances of VGG19, Inception-v3 and Xception models for all cases. It can be seen that the mix model performance at 40% - 60% of the real data mixed into the synthetic data, is close to that of the 70% mix model performance. This shows that the smaller datasets can add more generalization and performance when mixed and trained with synthetic data.

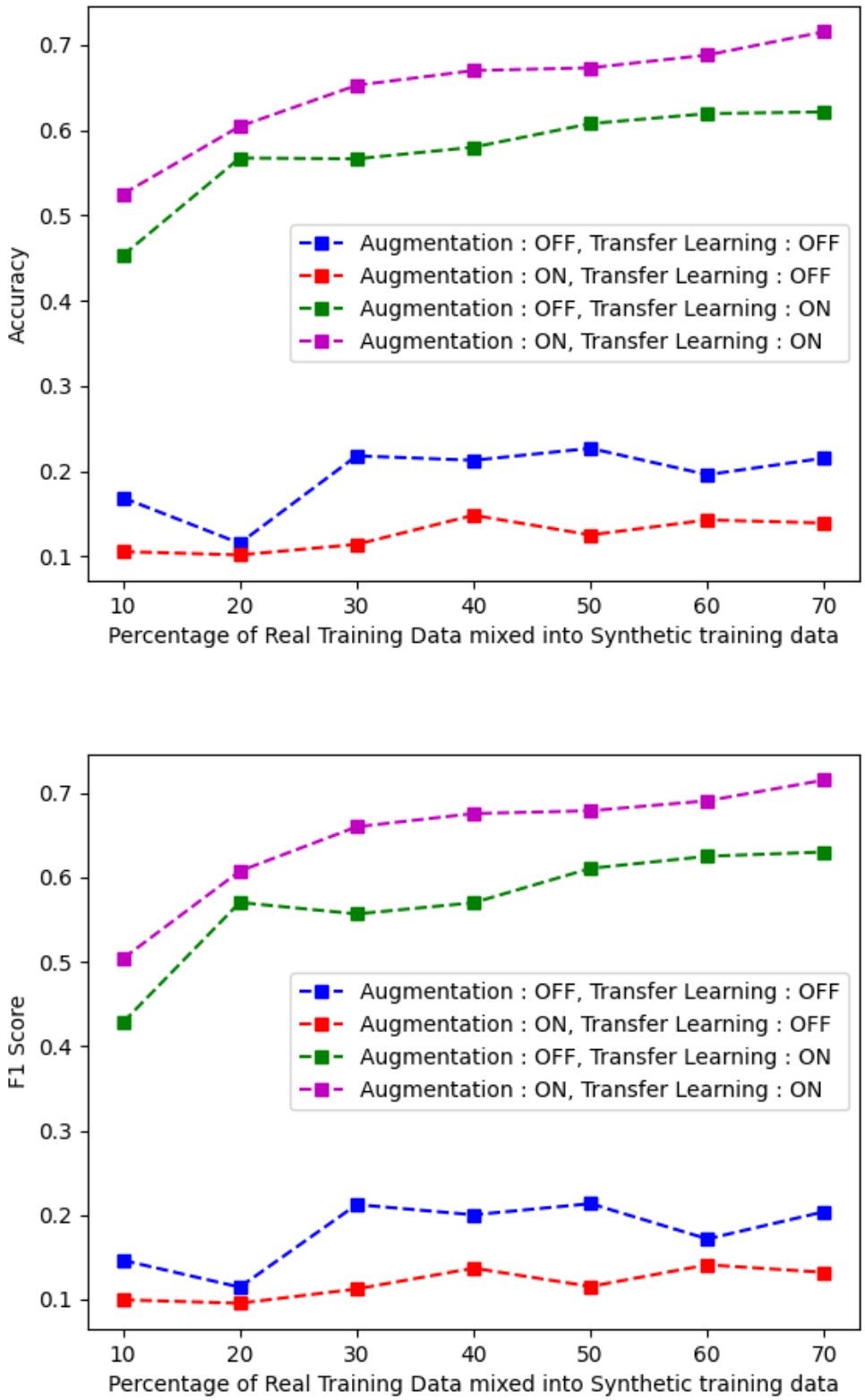

Fig. 6. comparison of Inception-v3 model performance in different cases of data augmentation and transfer learning (top) Accuracy (bottom) F1 score

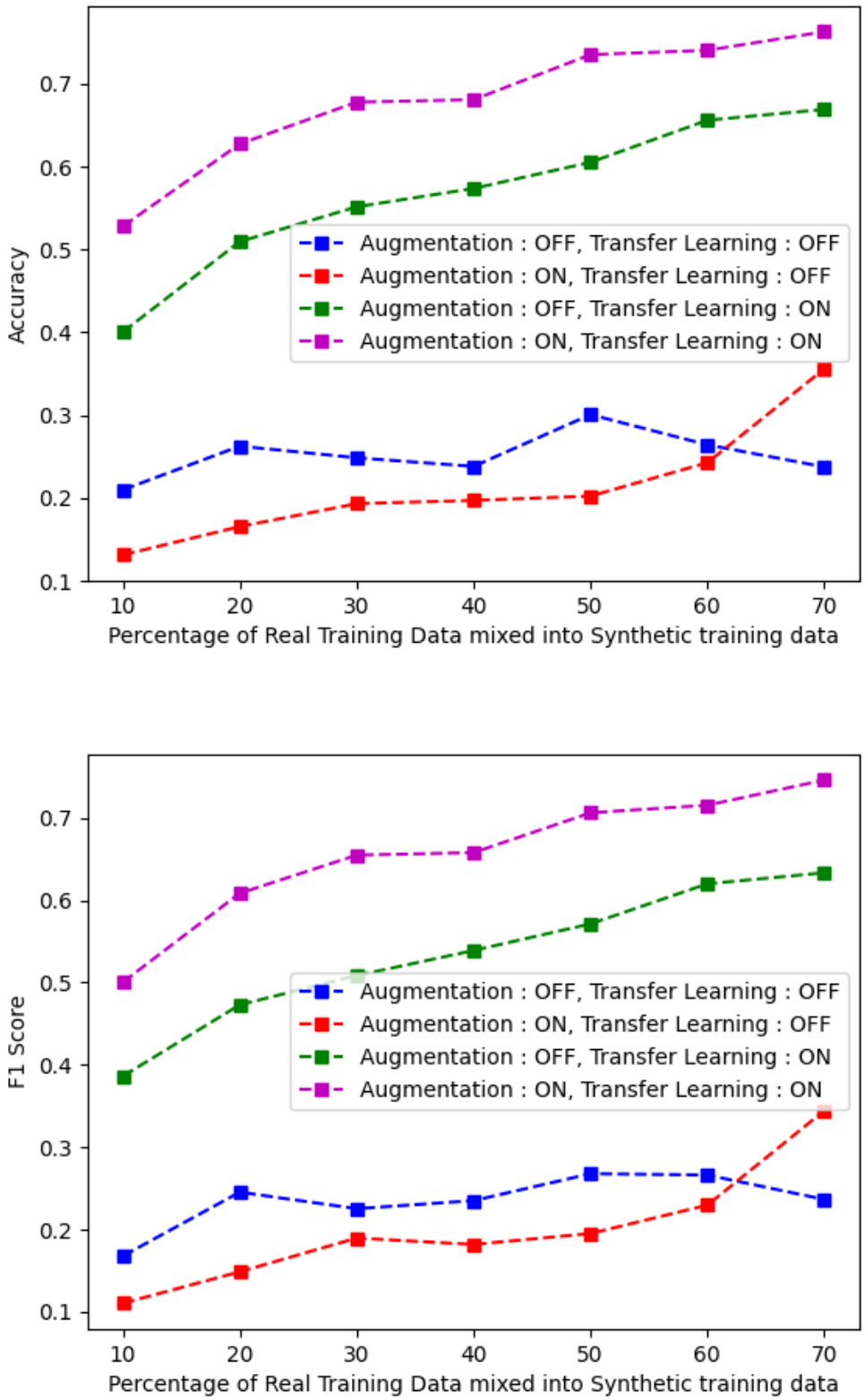

Fig. 7. comparison of Xception model performance in different cases of data augmentation and transfer learning (top) Accuracy (bottom) F1 score

Model	Real		Synthetic		Mix 10%		Mix 70%	
	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
VGG19	0.58	0.09	0.22	0.16	0.26	0.13	0.44	0.16
Inception-v3	0.74	0.74	0.25	0.24	0.45	0.43	0.62	0.63
Xception	0.80	0.78	0.28	0.25	0.40	0.39	0.67	0.63

Table 3. Model performance comparison with Transfer Learning and without Data Augmentation

Model	Real		Synthetic		Mix 10%		Mix 70%	
	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score	Accuracy	F1 score
VGG19	0.88	0.89	0.15	0.14	0.56	0.56	0.75	0.72
Inception-v3	0.09	0.0	0.08	0.03	0.17	0.15	0.22	0.20
Xception	0.09	0.0	0.09	0.08	0.21	0.17	0.24	0.24

Table 4. Model performance comparison without data augmentation and model weights trained from scratch

Tables 2-4 show the model performances on the real test set when trained with the synthetic dataset and with 10% and 70% mix ratios with real dataset with combinations of augmentation and transfer learning.

An interesting case to observe was that both Inception-v3 and Xception models performed very poorly even when trained with real datasets, even when all the weights were trained from scratch. Interestingly, the corresponding models trained with mixed datasets performed much better than that of models trained with real data in the same case.

In all cases, it has been found that training the model with synthetic train set alone yielded poor results compared to that of the models trained with real train set. However, even in the case of 10% mix ratio, the model performance improves significantly. As expected, with a higher mixed ratio, the performance is closer to that of the real dataset.It has been observed that while both transfer learning and data augmentation has positive effects on model performance, the effect of transfer learning is much more significant.

5 Conclusion

In this paper, a performance comparison of various models was demonstrated against the digits dataset and its real counterpart. While keeping the optimization parameters constant, the effect of data augmentation and transfer learning varied for each comparison. It has been shown that training with only the synthetic dataset results in poor performance on the real test set. However, transfer learning with imagenet weights has been shown to greatly improve the model performance trained on just synthetic dataset. Yet, at the same time, data augmentation has not been shown to improve the performance in this case. Most of the performance differences can be attributed to the reality gap that can be observed between the synthetic and real datasets. The performance difference can also be attributed to the images in synthetic dataset not being photo-realistic. Better performance may be achieved if more realistic models are used to build the synthetic datasets.

We have also discussed the possibility of addressing the reality gap by mixing various ratios of real data into our dataset and evaluating the model performances. Results show that real data mixing causes a significant performance increase compared to training the models with only synthetic datasets. Moreover, a higher mix ratio correlates to model performances close to that of the models trained with real datasets. However, acceptable performance results were achieved even when close to 50% of the real data was mixed

into the synthetic data for training, showing the positive effects of using this method with smaller datasets.

We can conclude that the models trained with our synthetic dataset are not suitable for real-world applications directly. We have shown that training models with real datasets mixed into the synthetic dataset makes them suitable for real-world applications. Thus, our method of building synthetic hand gesture datasets is shown to be very useful in cases where the existing real datasets are considerably small to generalize the real-world problem. Mixing the synthetic dataset with smaller datasets can help with building models that can better generalize images when deployed in the real world.

References

1. Chafic Abou Akar, Jimmy Tekli, Daniel Jess, Mario Khoury, Marc Kamradt, and Michael Guthe. Synthetic object recognition dataset for industries. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 150–155, 2022.
2. William S. Armstrong, Spencer Drakontaidis, and Nicholas Lui. Synthetic data for semantic image segmentation of imagery of unmanned spacecraft, 2022.
3. Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1841–1850, 2019.
4. François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
5. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
6. Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 2015.
7. Pranav Vaidik Dhulipala, Samuel Oncken, Steven Claypool, and Stavros Kalafatis. Synthetic datasets for hand gesture recognition. In *Proceedings of IEMTRONICS 2024: International IoT, Electronics and Mechatronics Conference, Volume 1*, chapter 3. Springer, 2024.
8. Pranav Vaidik Dhulipala, Samuel Oncken, Steven Claypool, and Stavros Kalafatis. Synthetic Hand Gesture Datasets: Digits and ASL, 2024. Dataset available at IEEE Dataport `https://dx.doi.org/10.21227/fje6-1t56`.
9. Huanzhang Dou, Wenhu Zhang, Pengyi Zhang, Yuhan Zhao, Songyuan Li, Zequn Qin, Fei Wu, Lin Dong, and Xi Li. Versatilegait: a large-scale synthetic gait dataset with fine-grainedattributes and complicated scenarios. *arXiv preprint arXiv:2101.01394*, 2021.
10. Salehe Erfanian Ebadi, You-Cyuan Jhang, Alex Zook, Saurav Dhakad, Adam Crespi, Pete Parisi, Steve Borkman, Jonathan Hogins, and Sujoy Ganguly. Peoplesanspeople: A synthetic data generator for human-centric computer vision. 2021.
11. Amr Gomaa, Robin Zitt, Guillermo Reyes, and Antonio Krüger. Synthogestures: A novel framework for synthetic dynamic hand gesture generation for driving scenarios. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–3, 2023.
12. Antonio Gulli. *Deep Learning Illustrated.* 2019.
13. Frederik Hagelskjær and Anders Glent Buch. Bridging the reality gap for pose estimation networks using sensor-based domain randomization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 935–944, 2021.
14. Boyong He, Xianjiang Li, Bo Huang, Enhui Gu, Weijie Guo, and Liaoni Wu. Unityship: a large-scale synthetic dataset for ship recognition in aerial images. *Remote Sensing*, 13(24):4999, 2021.
15. Dániel Horváth, Gábor Erdős, Zoltán Istenes, Tomáš Horváth, and Sándor Földi. Object detection using sim2real domain randomization for robotic applications. *IEEE Transactions on Robotics*, 39(2):1225–1243, 2022.
16. Mona Jalal, Josef Spjut, Ben Boudaoud, and Margrit Betke. Sidod: A synthetic image dataset for 3d object pose recognition with distractors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
17. Indu Joshi, Marcel Grimmer, Christian Rathgeb, Christoph Busch, Francois Bremond, and Antitza Dantcheva. Synthetic data in human analysis: A survey. *arXiv preprint arXiv:2208.09191*, 2022.
18. Muhammad Khalid. Sign language for numbers, 2019.
19. Murat Kuzhahiev and Yucel Cakir. A survey of public datasets for computer vision tasks. *Computer Vision and Image Understanding*, 2020.

20. Sigurd Kvalsvik, Ingeborg Rasmussen, Daniel Hagen, Teodor Nilsen Aune, and Per-Arne Andersen. Synthetic data generated in unreal engine 4, 2022.
21. Kyle Lindgren, Niveditha Kalavakonda, David E Caballero, Kevin Huang, and Blake Hannaford. Learned hand gesture classification through synthetically generated training samples. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3937–3942. IEEE, 2018.
22. Kapil Londhe. American sign language dataset. https://www.kaggle.com/dsv/2184214, 2021.
23. Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.
24. Teppei Miura and Shinji Sako. Synslag: Synthetic sign language generator. In *Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–4, 2021.
25. Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021.
26. Cristina Nuzzi, Simone Pasinetti, Roberto Pagani, Gabriele Coffetti, and Giovanna Sansoni. Hands: an rgb-d dataset of static hand-gestures for human-robot interaction. *Data in Brief*, 35:106791, 2021.
27. Shiqi Ou, Yan Fu, and Zhidong Xue. A survey on hand pose estimation with wearable sensors and computer-vision-based methods. *Sensors*, 20(4):1074, 2020.
28. Munir Oudah, Ali Al-Naji, and Javaan Chahl. Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging*, 6(8):73, 2020.
29. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.
30. Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11), 2021.
31. Heinrich Peters, Alireza Hashemi, and James Rae. Generalizable error modeling for human data annotation: Evidence from an industry-scale search data annotation program. *Machine Learning Research*, 2024. Accessed: 2024-11-09.
32. Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 145–151, 2020.
33. Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5296–5305, 2017.
34. Juergen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 2015.
35. Ramesh Shah and Neelam Gupta. Challenges in web scraping for machine learning: Data annotation and error correction. *Journal of Machine Learning and Data Mining*, 12(3):45–56, 2020.
36. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
37. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
38. Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30. IEEE, 2017.
39. Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 969–977, 2018.
40. Unity Technologies. Unity Perception package. `https://github.com/Unity-Technologies/com.unity.perception`, 2020. Accessed: 2022-11-09.
41. Svetozar Zarko Valtchev and Jianhong Wu. Domain randomization for neural network classification. *Journal of big Data*, 8(1):94, 2021.
42. Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017.
43. Steven Euijong Whang, Yuji Roh, Hwanjun Song, and Jae-Gil Lee. Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, 32(4):791–813, 2023.

Authors

Pranav Vaidik Dhulipala is currently a doctoral student in the Department of Electrical and Computer Engineering at Texas A&M University, College Station. He also received his Master of Science in Electrical Engineering from the same. He received his Bachelor's at Indian Institute of Technology Madras, India. His research interests include Computer Vision and Robotics.

Samuel Oncken received his Bachelor degree from Electrical and Computer Engineering Department from Texas A&M University, College Station.

Steven Clypool received his Bachelor degree from Electrical and Computer Engineering Department from Texas A&M University, College Station.

Stavros Kalafatis is Professor in the Department of Electrical and Computer Engineering at Texas A&M University, College Station. He is also the Associate Department Head of the Electrical and Computer Engineering Department. His research interests include Robotics, Aritificial Intelligence, Extended Reality (XR), and Computer Architecture.