Boosting Self-supervised Video-based Human Action Recognition Through Knowledge Distillation

Fernando Camarena * Leonardo Chang Miguel Gonzalez-Mendoza
Neil Hernandez-Gress
Tecnologico de Monterrey, School of Engineering and Science

Abstract

Deep learning architectures lead the state-of-the-art in several computer vision, natural language processing, and reinforcement learning tasks due to their ability to extract multi-level representations without human engineering. The model's performance is affected by the amount of labeled data used in training. Hence, novel approaches like self-supervised learning (SSL) extract the supervisory signal using unlabeled data. Although SSL reduces the dependency on human annotations, there are still two main drawbacks. First, high-computational resources are required to train a large-scale model from scratch. Second, knowledge from an SSL model is commonly finetuning to a target model, which forces them to share the same parameters and architecture and make it task-dependent. This paper explores how SSL benefits from knowledge distillation in constructing an efficient and non-task-dependent training framework. The experimental design compared the training process of an SSL algorithm trained from scratch and boosted by knowledge distillation in a teacher-student paradigm using the video-based human action recognition dataset UCF101. Results show that knowledge distillation accelerates the convergence of a network and removes the reliance on model architectures.

1 Introduction

Due to their classification performance and ability to extract multi-level representations without human engineering [6], deep neural networks (DNNs) are considered state-of-the-art in various computer vision [8, 21, 1], natural language processing [8, 19], and reinforcement learning tasks [18, 17],

Nevertheless, the model's performance is affected by two factors [31, 13]. On the one hand, DNNs involve sophisticated architecture designs [31], leading to over-parameterized models requiring extensive computational resources [31]. GPT-3 [15], one of the achievements in natural language processing models, consists of 175 billion parameters and is projected to require 3.14E23 FLOPS of computing. Using a V100 GPU will take 355 GPU years and cost around 4.6 million [15].

On the other hand, DNNs are usually trained in a supervised manner, requiring numerous high-quality labels [13]. Building a labeled dataset is an expensive process [13] that implies defining labeling manuals, class categories, storage pipelines, and labeling each observation by itself or by hiring an annotation service. For example, for a person, replicating the labeling process of ImageNet [7], composed of 14 million observations, will take 22 years [7]. On the other hand, people upload about 1 billion pictures and 300 hours of video to Facebook and YouTube daily, making it impossible to label.

A promising approach is self-supervised learning (SSL) [13], a novel learning schema that provides natural supervision using unlabeled data without human engineering. Nevertheless, SSL models still

*Corresponding author: fernando@camarenat.com, ORCID: 0000-0003-0888-2098

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

require high computing resources to train large-scale models [31, 34]. Second, feature representations learned by SSL are transferred to a target model using a finetuning methodology [9, 31], which exploits architecture-specific cues and, therefore, forces the models to share the same architecture's design and transfers only a facet of the SSL knowledge [34, 9].

Recent improvements [34] in the image and natural language processing suggest that knowledge distillation (KD) [9, 14, 34] can improve the efficiency of self-supervised methods. However, its application still needs to be clarified for video tasks [31].

This paper explores how SSL benefits from knowledge distillation in constructing an efficient and non-task-dependent training framework in a video-based human action recognition task. The experiments consist of three parts. First, we establish a baseline to assess the possible improvements in the model training and serve as the teacher models for the student's networks. The baseline are formed with C3D [34], R3D [28], and R(2+1)D [29] architectures, trained using the PCL [27] framework. Second, we train a student network with identical parameters as its teacher counterpart using KD to guide the student learning process. Third, we train the student model in a multi-architecture design configuration.

Our results show that knowledge distillation accelerates network convergence and removes the constraint of using the same architectural design. Providing flexibility in model target construction based on the application domain.

We divide the rest of this document as follows: We define the theoretical framework in section 2. Then, in Section 3, we introduce our proposal and define the experimental design. Next, in Section 4, we discuss the experimental design results. Finally, we provide our conclusions and future work in section 5.

2 Related Work

2.1 What is a human action?

To better comprehend the idea behind an action, picture the image of a person greeting another. Probably, the mental image formed involves the well-known waving hand movement. Likewise, if we create a picture of a man sprinting, we may construct a more dynamic image by focusing on the person's legs, as shown in Fig. 1. We unconsciously associate a particular message with a sequence of movements. This encoded sequence of gestures is what we will call "an action" [2]. The human action recognition goal is to build approaches that can understand the encoded message in the sequence.

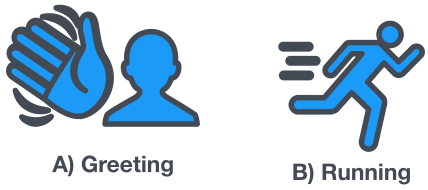

Figure 1: We instinctively associate a sequence of gestures with an action. For example, when we think of the action greeting, we might think of the typical hand wave. On the contrary, imagining a person running will create a more dynamic scene with movement centered on the legs. An action can be defined as a sequence of gestures that encode a message.

Handcrafted [35, 3, 4] and feature-learned [11, 36] methods are the two main approaches to recognizing human actions in the video. On the one hand, handcrafted [35, 3, 4] approaches entail manually engineering features [6], i.e., we must develop characteristics that support a computer to understand the human action concept. On the other hand, feature learning [36] methods extract multi-level representations without human engineering [6], which outperforms the performance of handcrafted methods. A popular setup is two-stream networks [24]. The concept is simple. On the one hand, a network extracts spatial characteristics from RGB images. On the other hand, a parallel network extracts motion information from optical flow information. Carretera et al. [5]

introduced the kinetics dataset as the foundation for re-evaluating the state-of-the-art architectures and the knowledge acquired to propose Two-Stream Inflated 3D ConvNet (I3D). I3D [5] demonstrates that 3D convolutional networks can be pretrained. Pretraining is a common practice for reducing processing time and the number of labeled data [26]. As a result, the concept of 2D CNN inflation was further researched [32, 20], resulting in novel architectures such as R(2+1)D [29].

2.2 Learning schemes

Video-based human action recognition approaches usually work on a supervised methodology [13], where the training algorithm employs labeled samples. Each label is usually annotated manually. Nevertheless, label annotation is a manual process that makes it expensive to process a high-dimensional dataset [34]. Consequently, reducing the dependency on labeling emerges as a research direction [12]. Self-supervised learning (SSL) [12, 13] relies only on unlabeled data to provide natural supervision. Its goal is to extract visual features whose performance is equal to or better than their supervised counterparts. SSL approaches are divided into two main branches [27]: Pretext tasks and contrastive learning [16]. On the one hand, pretext tasks define a classification function to understand the intrinsic nature of data samples [27]. Defining an auxiliary function is a challenging problem with ongoing developments. Some examples include a network if a set of video-frame has a consistent temporary line [10] or asking the network which transformation techniques were applied to an input frame [27].

On the other hand, contrastive learning [16] identifies what aspects of a video sample make it different from other samples. The core idea is to train a network to identify if two pairs of video features were extracted from the same video distribution. In this perspective, both types are complementary, as suggested in Pretext Contrastive Learning (PCL) [27]. PCL [27] is a joint optimization framework that uses a pretext task function to capture the local information and uses contrastive learning loss functions to gain a global view.

2.3 Knowledge Transfer

Label annotation is expensive and only practical for some application domains [13]. Hence, sharing the model knowledge is essential to reduce label dependency. Transfer learning [33] and finetuning [23] are the standard methods for transferring knowledge from one model to another, and they leverage the multi-level representations learned from deep learning architectures. The key idea is that some class objects share low-level features and can be used to re-construct a comparable entity [23]. The workflow of transfer learning and finetuning [23] consists of adding trainable layers on top of a model [33]. However, the transfers become architecture-dependent and, consequently, task-dependent [34].

Recently, knowledge distillation [34, 31, 9] served as the foundation of a novel approach to transfer learning. In contrast to the finetuning [23] and transfer learning techniques, KD [9] does not build on top of the pretrained model. Instead, KD [9] defines a teacher and student framework where the teacher uses the model's outputs as guidance.

KD has previously been explored in the literature [34, 31, 9, 22, 30], achieving remarkable performance. [22] explores how knowledge distillation serves as auxiliary supervision to efficiently learn larger pretrained language models. SSKD [34], which uses KD to extract richer dark knowledge, is proposed to improve image classification performance. In [30], they explore how KD affects the video action recognition task in a self-distillation approach. Our work differs from [30] because we explore how KD affects a self-supervised approach inside an action recognition, whereas [30] works under a supervised methodology. Also, [30] proposed self-distillation methods that use previous versions of the teacher. Our method aims to leverage what the community has already trained and use it in a traditional KD approach.

3 Proposed Work

This document aims to study how knowledge distillation affects a self-supervised methodology in the human action recognition video task. We are interested in how the convergence of the model and classification performance is affected using multiple architecture designs.

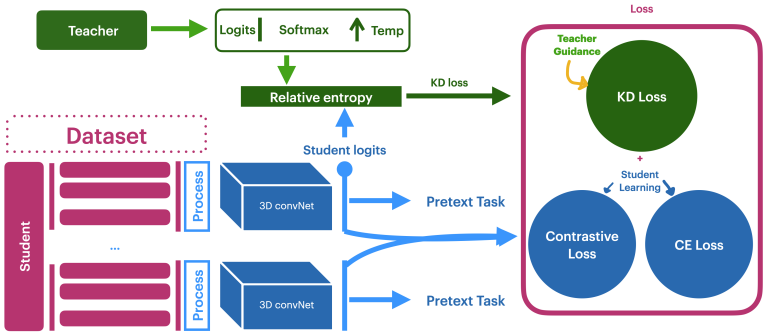

Figure 2: Knowledge distillation and self-supervised learning framework for video tasks. The loss function for the student will be composed of both self-training using the PCL scheme and the guidance of a master model using the relative entropy between the outputs of the two models.

The intuition is that the learning capabilities of humans are not limited to self-taught. Instead, since we are children, our training has been surrounded by various teachers who filter knowledge and provide it in a way that facilitates knowledge assimilation. In the deep learning community, we have already trained many models. Nevertheless, we usually train a new model from scratch or use transfer learning techniques that, as previously described, force the target model to share the same architectural design.

To transfer knowledge, we follow a traditional KD teacher-student framework [34, 9], described in Fig. 2. The key idea is to learn through imitation by requesting the student network to mimic the teacher's network probabilities. To increase the information students can learn, we scale the softmax probabilities using a temperature value to remove probabilities near zero. The output probabilities are compared using the Kullback-Leiber divergence [9], also known as relative entropy.

We train the SSL models using the PCL framework, with the pretext task being "which transformation was done to the input video" using the cross-entropy function. Furthermore, we use a perceptron as the projection head for contrastive loss.

We tested the widely used C3D [34], R3D [28], and R(2+1)D [29] architectures. Testing on multiple video architectures reduces the bias in the experiments and, therefore, gives a firm notion about the effectiveness of the knowledge transfer. Furthermore, employing several architectures allows us to reduce the influence of the learning rate bias on model convergence caused by the diverse number of parameters used in tested architectures.

To provide flexibility and help the student outperforms the teacher model [34], in addition to the teacher guidance, we add both the pretext and the contrastive loss in the training algorithm, as shown in Fig. 2,

We conducted the experimental design using the UCF101 [25] dataset, a benchmark dataset in action recognition tasks. This dataset consists of 13,320 videos collected from YouTube and divided into 101 categories.

The experiments consist of three parts. First, we establish an SSL baseline and then transfer between the same architecture and different architectures. Let us break down each experiment set; the first one aims to establish a baseline to assess the possible improvements in the model training and serves as the teacher models for the student's networks. The baseline are formed with C3D [34], R3D [28], and R(2+1)D [29] architectures, trained using the PCL [27] framework with the same parameters over 100 epochs.

The second and third experiments investigate how KD influences SSL. To begin, we train a student network with identical parameters as its teacher counterpart. This experiment determines whether KD is a viable way of knowledge transfer. Second, we train the student model with different architecture designs to its teacher model to demonstrate that KD is not model agnostic. We

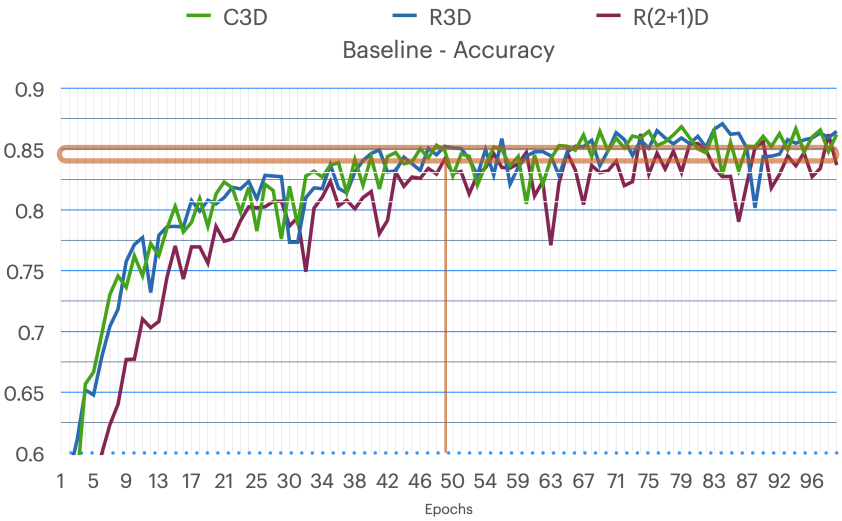

Figure 3: Results from our baseline, three architectures were tested using a PCL-based self-supervised approach. All architectures have similar performance.

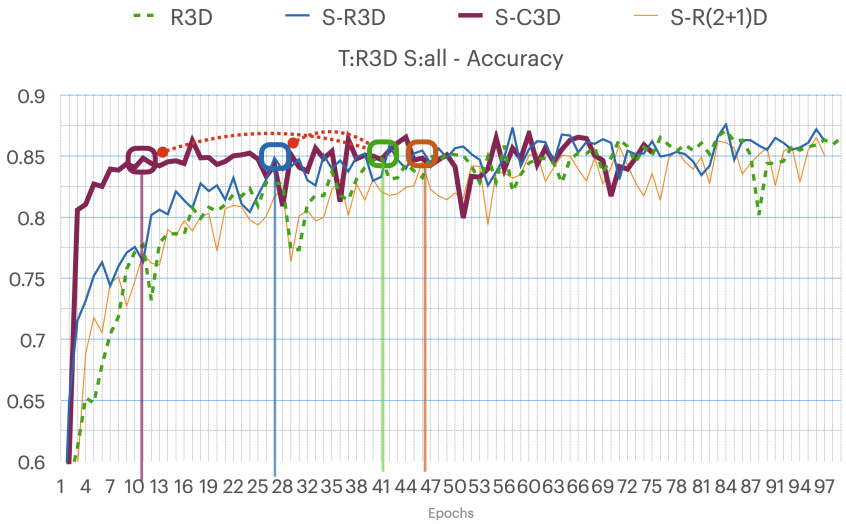

Figure 4: Accuracy plot for the training of the student model using the same architecture as its teacher (R3D). It is observed that the learner model converges faster than the regular model.

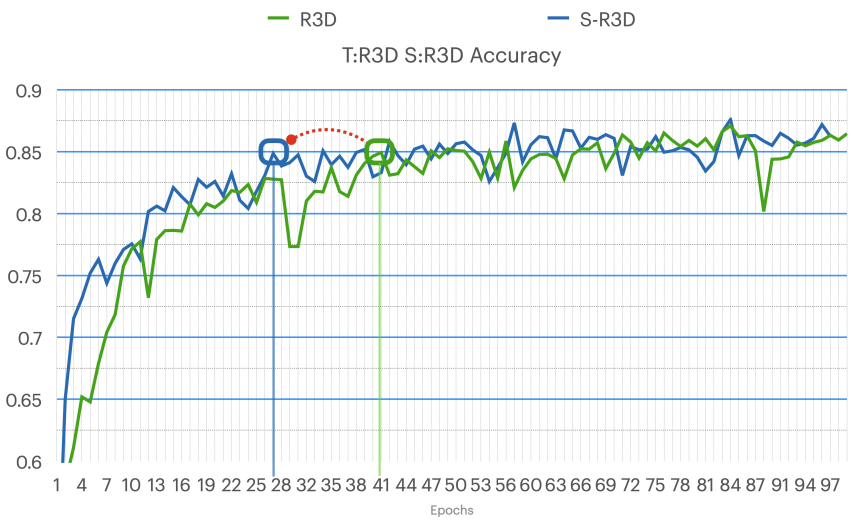

Figure 5: Accuracy plot for the training of the student model using the same as its teacher (R3D). It is observed that the learner model converges faster than the regular model.

4 Preliminary Results and discussions

Fig. 3 compares the accuracy performance of the C3D [34], R3D [28], and R(2+1)D [29] backbones on the UCF101 [25] dataset. All the architectures have a similar performance and achieve about 85 percent of classification accuracy by epoch 50. Consequently, to consider that KD benefits the SSL training process, it should get a higher accuracy or achieve similar performance in a lower epoch number.

Fig. 4 describes the accuracy of the student model compared to its teacher counterpart using the R3D [28] backbone. Relevant points include that the student converges faster and outperforms the teacher's accuracy in almost half of the epochs. Hence, in the case of the R3D architecture, KD boosts the SSL training process. On the other hand, Fig. 5 describes the accuracy performance using the C3D [34] architecture; like the case of the R3D [28] backbone, the student network outperforms its teacher counterpart in both convergence and classification performance. Furthermore, the insights are consistent with the R(2+1)D [29] architecture, shown in Fig. 6.

All architectures presented show that using the guidance of pretrained models helps train SSL models faster without sacrificing performance.

Nevertheless, researchers continuously propose more and more architectures, and the ability to transfer knowledge using different architectural settings and reuse old knowledge learned in novel settings is crucial to preserving information. Also, when using different architectural settings, the transfer is not based on architectural cues but forces the model to transfer the feature representation of the objects. Finally, different architectural designs enable the training of lower-size models to suit low-computational power device requirements. In contrast, reusing knowledge to create larger models is also viable.

Fig. 7 shows the accuracy performance using R3D [28] as the selected architecture for the teacher and C3D [34], R3D [28], and R(2+1)D [29] for the students. There are two main points from Fig. 7. First, the student networks get comparable performance to their teacher counterpart, enabling the transfer between different settings. Second, the best model uses different architecture designs; in this case, using an R3D [28] for the teacher model and a student using C3D [34]. On the other hand, Fig. 8 describes the model's performance using C3D [34] for the teacher model and C3D [34], R3D [28], and R(2+1)D [29] for the students; the figure shows a similar pattern to the R3D [28] architecture; students tend to outperform and out-converge the master models. Also, in the same case of Fig. 7, using KD in an SSL setting enables the knowledge transfer between different architectures settings. Finally, Fig. 9 illustrates the accuracy performance of the teacher model using the R(2+1)D [29] architecture and C3D [34], R3D [28], and R(2+1)D [29] for the students. The results are consistent with previous architectures; all the student models converge faster than the

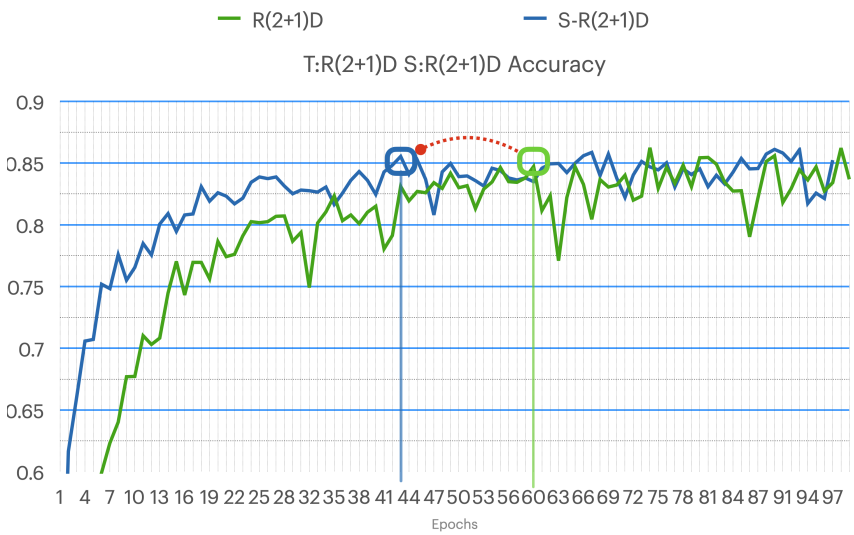

Figure 6: Accuracy plot for the training of the student model using the same architecture as its teacher (R(2+1)D). It is observed that the learner model converges faster than the regular model.

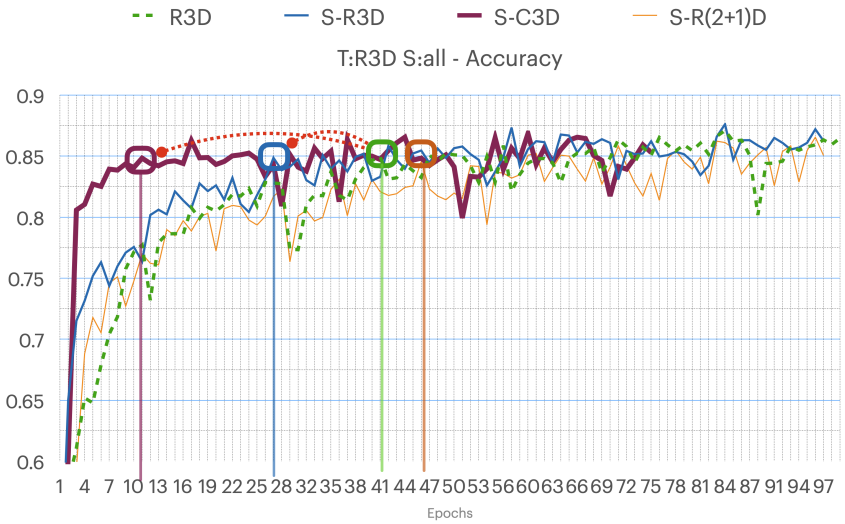

Figure 7: Accuracy plot for the training of the student model using different architectures as its teacher (R3D). All the student models have similar performance to the teacher model. It is possible to transfer knowledge using different architectures and even outperform the model training using the same configuration settings.

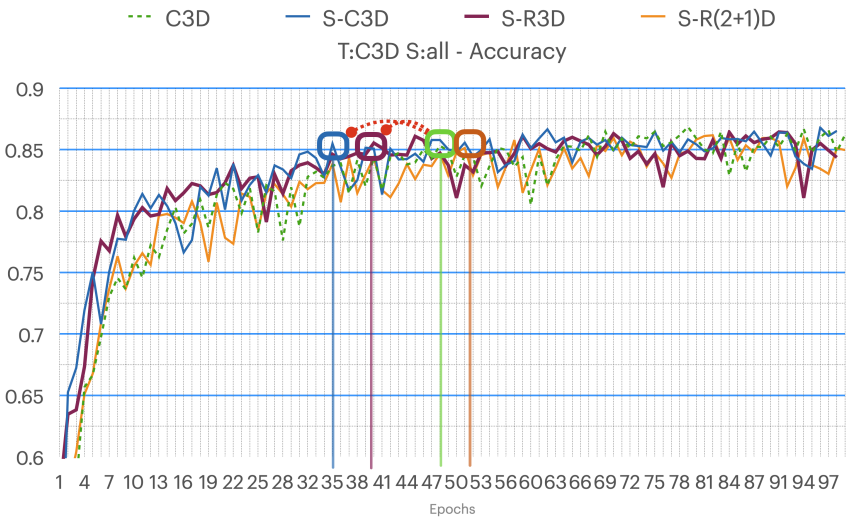

Figure 8: Accuracy plot for the training of the student model using different architectures to the teacher (C3D). All the student models have similar performance to the teacher model. It is possible to transfer knowledge using different architectures and even outperform the model training using the same configuration settings.

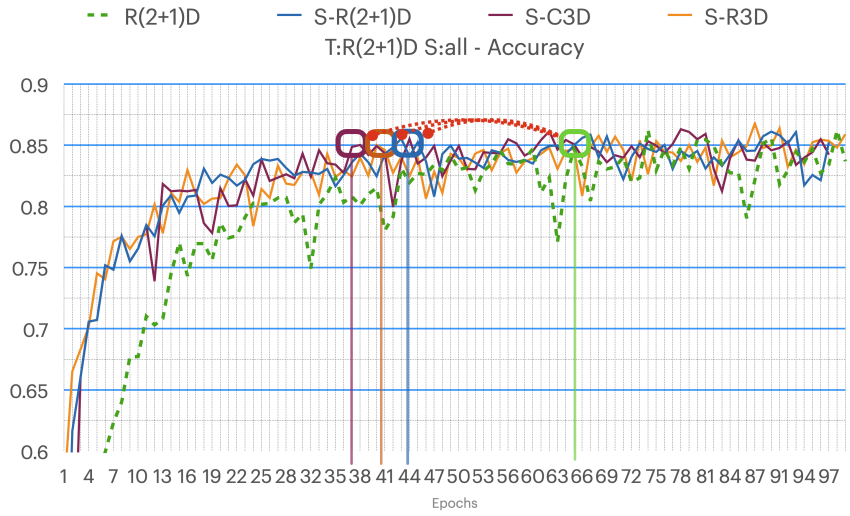

Figure 9: Accuracy plot for the training of the student model using different architectures as its teacher (R(2+1)D). All the student models have similar performance to the teacher model. It is possible to transfer knowledge using different architectures and even outperform the model training using the same configuration settings.

master model, and transferring knowledge using different architectures is feasible with little-to-none impact on the classification accuracy.

5 Conclusion

This work explores how self-supervised learning can benefit from knowledge distillation in terms of convergence, flexibility in architectural design, and performance. Preliminary experiments focus on testing the effects of knowledge distillation in self-supervised learning using video-based human action recognition as the application domain. The central insight found:

- KD is a viable option to transfer knowledge from one model to another for the action recognition task in a self-supervised framework.
- KD helps a self-supervised model to converge faster for video-action recognition.
- Even if the architecture and hyperparameters are different, it is possible to move knowledge from a teacher to a student model. A crucial aspect of preserving knowledge. Since researchers continuously propose more and more architectures. Another advantage is to create models that fit the application's constraints better. For example, creating lower-size models to suit lower-computational power devices or larger models when resources are not a problem.
- KD provides flexibility in the configuration design that helps students outperform the teacher model's classification performance.

Future work considers new datasets, application domains, techniques to assess and visualize learned features, and novel approaches to distill knowledge.

References

[1] Claudine Badue, Rânik Guidolini, Raphael Vivacqua Carneiro, Pedro Azevedo, Vinicius B Cardoso, Avelino Forechi, Luan Jesus, Rodrigo Berriel, Thiago M Paixao, Filipe Mutz, et al. Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816, 2021.

[2] Paulo Vinicius Koerich Borges, Nicola Conci, and Andrea Cavallaro. Video-based human behavior understanding: A survey. *IEEE transactions on circuits and systems for video technology*, 23(11):1993–2008, 2013.

[3] Fernando Camarena, Leonardo Chang, and Miguel Gonzalez-Mendoza. Improving the dense trajectories approach towards efficient recognition of simple human activities. In *2019 7th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2019.

[4] Fernando Camarena, Leonardo Chang, Miguel Gonzalez-Mendoza, and Ricardo J Cuevas-Ascencio. Action recognition by key trajectories. *Pattern Analysis and Applications*, 25(2):409–423, 2022.

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[6] OpenCV AI Courses. Deep learning with PyTorch, 1 2020.

[7] 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.

[8] Omar Elharrouss, Noor Almaadeed, Somaya Al-Maadeed, Ahmed Bouridane, and Azeddine Beghdadi. A combined multiple action recognition and summarization for surveillance video sequences. *Applied Intelligence*, 51(2):690–712, 2021.

[9] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge Distillation: A Survey. *arXiv*, 2020.

[10] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5548–5558, 2021.

[11] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017.

[12] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[13] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):4037–4058, 2020.

[14] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammus: A survey of transformer-based pretrained models in natural language processing. *arXiv preprint arXiv:2108.05542*, 2021.

[15] Lambda Labs. OpenAI's GPT-3 Language Model: A Technical Overview, 6 2020.

[16] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[17] Chen Lei. Deep reinforcement learning. In *Deep Learning and Practice with MindSpore*, pages 217–243. Springer, 2021.

[18] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[19] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021.

[20] Guocheng Liu, Caixia Zhang, Qingyang Xu, Ruoshi Cheng, Yong Song, Xianfeng Yuan, and Jie Sun. I3d-shufflenet based human action recognition. *Algorithms*, 13(11):301, 2020.

[21] Manuel Martinez, Lukas Rybok, and Rainer Stiefelhagen. Action recognition in bed using bams for assisted living and elderly care. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 329–332. IEEE, 2015.

[22] Yujia Qin, Yankai Lin, Jing Yi, Jiajie Zhang, Xu Han, Zhengyan Zhang, Yusheng Su, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Knowledge Inheritance for Pre-trained Language Models. *arXiv*, 2021.

[23] Ricardo Ribani and Mauricio Marengoni. A survey of transfer learning for convolutional neural networks. In *2019 32nd SIBGRAPI conference on graphics, patterns and images tutorials (SIBGRAPI-T)*, pages 47–57. IEEE, 2019.

[24] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[25] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[26] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Pretext-Contrastive Learning: Toward Good Practices in Self-supervised Video Representation Leaning. *arXiv*, 2020.

[27] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Selfsupervised video representation using pretext-contrastive learning. *arXiv preprint arXiv:2010.15464*, 2:2, 2020.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[29] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[30] Duc-Quang Vu, Ngan Le, and Jia-Ching Wang. Teaching yourself: A self-knowledge distillation approach to action recognition. *IEEE Access*, 9:105711–105723, 2021.

[31] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[32] Xianyuan Wang, Zhenjiang Miao, Ruyi Zhang, and Shanshan Hao. I3d-lstm: A new model for human action recognition. In *IOP Conference Series: Materials Science and Engineering*, volume 569, page 032035. IOP Publishing, 2019.

[33] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[34] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *European Conference on Computer Vision*, pages 588–604. Springer, 2020.

[35] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. A review on human activity recognition using vision-based method. *Journal of healthcare engineering*, 2017, 2017.

[36] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.