Final report
Knowledge distillation for echocardiogram view classification

Andris Freimanis
gusandrifr@student.gu.se

Moritz Sprenger
gussprmo@student.gu.se

Raouf Bahsoun
gusbahsra@student.gu.se

Yu-Ping Hsu
gushsuyu@student.gu.se

November 2, 2023

Supervisors
Yinan Yu, academic supervisor, yinan@chalmers.se
Charlotte von Numers, industry supervisor, charlotte.vonnumers@astrazeneca.com
Luis Arevalo, industry supervisor, imarevost@gmail.com

Contents

1 Abstract **3**

2 Introduction **3**

3 Literature Review **4**

4 Methods **7**

5 Experimental setup **8**

6 Results **9**

7 Limitations **13**

8 Discussion **14**

9 Conclusion **15**

1 Abstract

The heart is responsible for promoting blood circulation, transporting oxygen and nutrients to various organs and tissues of the body, as well as carrying away waste and carbon dioxide from the body. Having a normal ejection fraction value is vital for human health. Echocardiography is the most common method of measuring ejection fraction but it is highly dependent on correct execution by the practitioner to get the desired views for diagnostic purposes. Four different ultrasound views are classified using four different model architectures. Knowledge distillation methods are used to extract the knowledge from an existing VGG16 model and transfer it to small student models. The purpose of this project is to find an optimal student model with high accuracy, fast classification speed, and small model size so that it can perform optimally on mobile devices to help classify views and guide the correct image scanning. The four student models studied are Mobilenet-V3-small, Mobilenet-V3-large, Shufflenet, and Ghostnet. The results show that MobileNet-V3-large had the highest relative validation accuracy at 0.98. MobileNet-V3-small and ShuffleNet, are two smaller models and can achieve higher inference speedups of up to 9. All student models keep over 0.95 of the teacher's performance, while being at least 30 times smaller in memory size.

2 Introduction

The heart is the key organ that maintains the normal operation of various functions of the human body. It controls the flow of blood. Without blood which supplies nutrients, organs, tissues or cells in the body are unable to continue working normally. The heart is located on the left side of the human chest and is divided into two parts, the left heart and the right heart. This division prevents the heart from mixing deoxygenated blood and oxygenated blood. Oxygenated blood is pumped by the left side of the heart, proceeds to systemic circulation via the aorta, and travels to various parts of the body to supply cell nutrients. After returning to the right atrium, it becomes hypoxic blood and travels to the lungs through the pulmonary artery to get oxygen. The oxygen-rich blood returns from the pulmonary veins to the left atrium, completing the pulmonary circulation and preparing for the next round of systemic circulation. The heart is like an electric pump, pumping blood throughout the body. The power of the pump mainly relies on the sinoatrial node located in the right atrium to send electrical signals to stimulate the contraction of myocardial tissue to generate heartbeats. Each heartbeat is a contraction and relaxation of the heart. The heart muscle contracts and relaxes regularly with each beat of the heart. When the heart contracts, blood is pumped from the left ventricle to the rest of the body. At the same time, the heart valves close to prevent blood from flowing backward. During relaxing, the valves open to allow blood to flow into and fill the ventricles in preparation for the next contraction. Ejection fraction is the percentage of blood pumped out of the heart each time it beats.

A healthy human heart can only pump half to two-thirds of the blood from the left ventricle with each contraction. This means the normal ejection fraction is not 100% for a healthy person. The normal ejection fraction is between 55% and 65%. The JACC: Heart Failure published in August 2022 reveals the impact of abnormal left atrial ejection fraction on the heart. Values that are too high or too low will increase the hazard ratio of major adverse

cardiovascular events. The ejection fraction can be measured by the following method: (1) Echocardiography: Echocardiography, known as cardiac ultrasound, uses sound waves to create images of the heart's beating. These images show how blood flows through the heart and its valves. (2) Cardiac magnetic resonance imaging (MRI): It uses magnetic fields, radio waves, and a computer to create transverse images of the heart. (3) Cardiac computed tomography (CT) scan: X-rays from different angles are used to construct images of the heart. (4) Nuclear medicine scan: During a cardiac nuclear medicine scan, a small amount of radioactive material is injected into the bloodstream through a vein. Single photon emission CT/CT (SPECT/CT) and positron emission tomography/CT (PET/CT) cameras are used to track the radioactive material in the blood as it flows through the heart and lungs. Because cardiac images from MRI, CT, and nuclear medicine are all static, echocardiography is the most common method of measuring ejection fraction [27].

The basic principle of cardiac ultrasound uses the pulse-echo principle. The ultrasound transducer converts electrical signals into ultrasonic pulses. The pulse passes through the skin and into the internal anatomy at different speeds. When the ultrasonic sound waves encounter two different tissues with different characters or densities, the wave reflects. This phenomenon is called ultrasound reflection. The transducer receives the returning echoes and converts them to electric signals which a computer converts into points of brightness on the image corresponding to the anatomic position and the strength of the reflecting echoes. A complete image frame formed by ultrasound is called a sonogram. When a patient is diagnosed by ultrasound, many sonograms will be reconstructed into a video.

The limitation of cardiac ultrasound is that the ultrasound wave will be obstructed by air or bones and cannot obtain clear images. For some patients with chest wall abnormalities, emphysema, or obesity, it is hard to sample images with good resolution. This makes diagnosis difficult. Cardiac ultrasound is an examination that highly relies on technology and experience. Providing a machine learning model on a mobile device or the medical equipment, that helps classify these views, could assist and guide operators in sampling correct images. This would greatly improve the imaging quality and further assist doctors in more accurate measurements of left ventricular ejection fraction. This way, the hidden dangers of the patient's heart can be detected earlier, or the effectiveness of the patient's heart treatment can be measured more accurately.

The purpose of this project is to use knowledge distillation to transfer the knowledge of an existing teacher model to a student model and find an optimal student model with high accuracy, fast classification speed, and small model size so that it can perform optimally on mobile devices, for instance on a Raspberry Pi computer, given the requirement of being able to produce a video at 30 frames per second and having a memory size of less than 50 Megabytes.

3 Literature Review

Knowledge distillation (KD) denotes a method where a smaller student model is supervised by a larger teacher model and was popularized by [12]. The idea behind KD is that the student model mimics the teacher model and reaches similar or even greater performance with less

capacity [8]. Figure 1 shows the general setup for knowledge distillation methods. Both models are trained with the same data and the student network is supported by knowledge transferred from the teacher model.

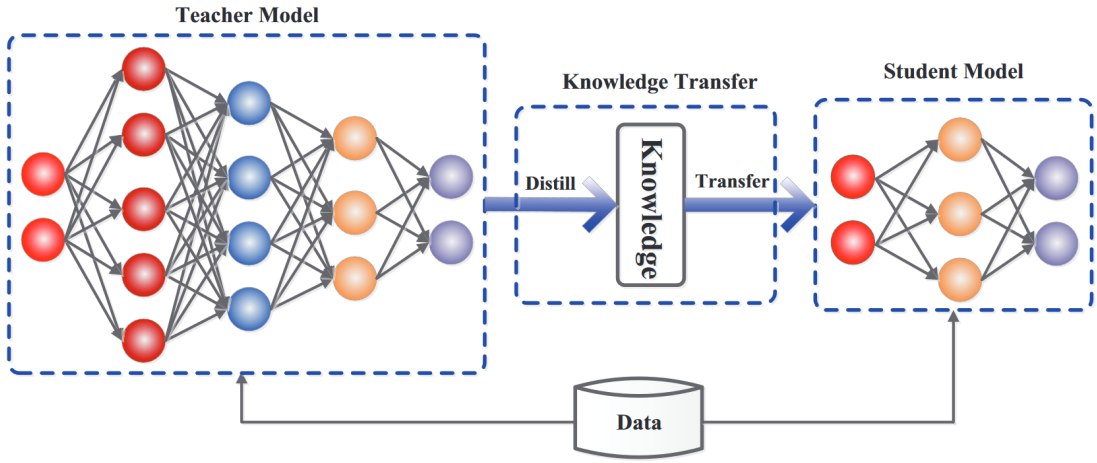

Figure 1: Basic framework for knowledge distillation [8]

Knowledge distillation system can generally be differentiated by their three main components: the type of knowledge, the distillation algorithm and the teacher-student architecture [8].

Three main categories of knowledge are used to describe KD systems: response-based knowledge, feature-based knowledge and relation-based knowledge [8].
Response-based knowledge refers to the outputs of the last layer in a DNN. This is the simplest type of knowledge transferred in KD systems [8]. The learning objective of the student network is to mimic the outputs of the teacher network. With labeled data available, this objective can be combined with the loss regarding the ground truth labels as a weighted sum in the loss function [12]. Using this type of knowledge is applicable in many domains. The most popular use is for (image) classification tasks, where the softmax outputs are used [12, 4]. Some works extend this approach by only using teacher outputs when the teacher classification is correct [19] or adapting the softmax function throughout the training process to improve the knowledge transfer [18]. Further tasks are object detection [6], pose estimation [31] or LLMs [10], which extend the distillation loss with domain specific outputs such as offsets or heatmaps.
Since DNNs learn through building more and more useful representations in their intermediate layers, feature-based knowledge used for distillation refers to outputs based on the intermediate layers of networks. The distillation loss is then computed by matching the activations or proxies in the teacher model with those in the student model [8]. Some work match the feature activations directly [5, 24], while others use proxies such as activation statistics [22] or other derived information [16]. For feature-based knowledge two main questions have to be explored, which intermediate layers to choose for distillation, since the number of layers in the teacher network might be larger, and how to match feature representations, when spatial dimensions differ between teacher and student [8]. For the former, some works use attention

to guide the choice of layers [5, 24]. The latter can for example be tackled by the usage of singular value decomposition [16].
While the discussed knowledge types consider singular specific layers of the teacher network, relation-based knowledge tries to capture information about the relationships between layers or data samples to guide the student network [8]. Works that focus on layer relationships use correlations between pairs of feature maps as knowledge [16] or try to model and match the information flow between teacher and student model [23]. The relations between data samples/instances is also used for knowledge distillation. Methods considering instance relations consider the spatial relations between multiple instances and try to match these structural relations between student and teacher models [21, 7].

Distillation algorithms can be grouped into three main categories: offline distillation, online distillation and self-distillation [8].
Most of the presented works use offline distillation, where the teacher model is trained first or assumed to be already trained. The student model is then trained in a second step under the guidance of the teacher model, which makes training more efficient [8].
In online distillation algorithms both teacher and student model are trained jointly in an end-to-end fashion, this can further improve the student performance and be necessary when no pretrained teacher model exists [8]. In deep mutual learning, multiple networks train jointly and each model functions as teacher and as a student model during the training process by transferring their response-based knowledge between them [34]. This approach is extended by [15], who fuses intermediate features used by a classifier that distributes its' knowledge back to the student networks.
In self-distillation the knowledge transfer is not between different networks but inside the same network. Some works distill knowledge from one section of a network to another, for example knowledge deeper layers of the network to shallower ones [32] or between different output layers in multi-exit architectures [26]. In other works the knowledge transfer happens between different times in the training process, where knowledge from earlier epochs is transferred to later epochs [30].

Next to the type of transferred knowledge and the distillation algorithm, the architecture of the teacher and student model also plays a role in knowledge distillation.
Due to the compression intention behind the KD approach most student networks are less complex than the teacher networks. This can either be achieved by student models being a simplified version of the teachers with less layers and/or channels [29, 17], a quantized network with the same structure [28] or other size optimized network architectures [14, 9]. In online distillation settings, teacher models can have the same architecture as the student or be an ensemble of them [34]. Due to the capacity gap between larger teacher and smaller student networks, the student networks might not be able to replicate the performance of teachers [20]. Some works tackle this problem by reducing the student capacity in a controlled manner [9] or by iteratively introducing smaller and smaller teacher assistants that are first acting as the student and then as the teacher for the next teacher assistant until the desired size of a final student model is reached.

4 Methods

Four ultrasound views are used to train the classification models which are A2C, A4C, PLAX, and PSAS. A2C is the apical two-chamber view. A4C is the apical four-chamber view. PLAX is parasternal Long-Axis view. PSAS is parasternal Short-axis view. When a patient is diagnosed by ultrasound, many sonograms are reconstructed into a video. Each video contains only one view. The videos were split into training, validation, and test sets. Each video in these sets was split into images. The videos used in this project come from three source datasets [1][2][3].

Given the described initial conditions for our project, we decided to focus on offline response-based knowledge distillation techniques and investigate the trade-off between performance and model size/speed. This can be explained by the availability of an already trained and well performing teacher network and the transferability of response-based implementations between different network structures. We identified three response-based knowledge distillation techniques suitable for the required task.
The knowledge of a network can be distilled by matching the *logits* and in the case of (multi-class) classification the softmax distribution of the teacher and student networks [12]. The distillation loss L_D can then be expressed as in equation 1, where $L(.)$ represents a loss function, $p(.)$ the softmax function and z_t, z_s the logits of teacher and student respectively [8].

$$L_D = L(p(z_t), p(z_s)) \tag{1}$$

For well performing networks the output softmax distribution often has a low entropy with much of the mass concentrated on the predicted class. To transfer more knowledge to students, a distribution with higher entropy, that contains more information about similarities between samples, can be desirable. Equation 2 shows an adapted softmax function with z_i as the logit for class i and T as the temperature, which increases the entropy of the resulting probability distribution and results in so called soft targets [12].

$$p(z_i, T) = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)} \tag{2}$$

Combining equations 1 and 2 leads to equation 3, the distillation loss with soft targets where L_{KL} is the Kullback Leibler divergence loss.

$$L_D(p(z_t, T), p(z_s, T)) = L_{KL}(p(z_t, T), p(z_s, T)) \tag{3}$$

To leverage the student's training data. the distillation loss can be combined with the student-/classification-loss L_S, which is the cross-entropy loss between student outputs and the ground truth labels $L_S = L_C E(y, p(z_s, T = 1))$. Combining these two losses, leads to a joint loss given in equation 4, where α is a balancing parameter between the two losses and the multiplication with T^2 is done to normalize the magnitude of the gradients [12]. Training with this loss function as an objective is further referred to as vanilla knowledge distillation.

$$L_J = \alpha(L_D * T^2) + (1 - \alpha)L_S \tag{4}$$

To avoid the heuristic tuning of the balancing hyperparameter α, Meng et al. suggest to simplify the joint loss L_J. They suggest to only use the distillation loss when the teacher network makes a correct prediction, but replace the softmax outputs with the one-hot encoded hard labels when the teacher makes an incorrect prediction. They call this approach conditional teacher student learning [19].

Curriculum learning is a technique, where the learning difficulty is increased through the training process [18]. In contrast to a static and heuristically chosen hyperparameter T in the introduced methods so far, Zheng et al. make use of a dynamic temperature T_dyn, that is dynamically adapted throughout the training process. The temperature is modeled as a temperature module that gets learned in an adversarial manner. The temperature module is optimized in the opposite direction of the student, trying to maximize the distillation loss with the update shown in equation 5, where θ refers to the temperature parameter and μ is the learning rate [18].

$$\theta_{temp} = \theta_{temp} + \mu \frac{\partial L}{\partial \theta_{temp}} \tag{5}$$

To adopt curriculum learning and change the difficulty of the task throughout the training, the loss w.r.t. the temperature is scaled by λ, leading to equation 6 [18].

$$\theta_{temp} = \theta_{temp} + \mu \frac{\partial(\lambda L)}{\partial \theta_{temp}} \tag{6}$$

λ increases with each epoch E_n during the training according to either a linear schedule or a cosine schedule given in equation 7, where λ_{min} and λ_{max} are the range for λ and E_{cap} is the epoch at which λ reaches it maximum and stays constant. The linear schedule also adopts the same methodology with E_{cap}.

$$\lambda_n = \lambda_{min} + \frac{1}{2}(\lambda_{max} - \lambda_{min})(1 + cos((1 + \frac{min(E_n, E_{cap})}{E_{cap}})\pi)) \tag{7}$$

Adopting their implementation of a global temperature, T is a single learnable parameter used for each sample in a batch before being updated with its gradient [18]. Following Zheng et al. we set λ_{min} and λ_{max} at 0 and 1 as well as E_{cap} at 10 as our default values.

5 Experimental setup

In accordance with the project goals an ablation study with 4 different student architectures and the 3 described distillation techniques is conducted. As for student architectures we focused on efficient convolutional architectures especially designed for mobile and edge devices. The four considered architectures are *Mobilenet-V3-small*, *Mobilenet-V3-large* [13], *Shufflenet* [33] and *Ghostnet* [11]. All network implementations are taken from PyTorch [25]. To confirm with the expected input sizes of these architectures, the greyscale image data is duplicated along the channel axis. Additionally, images are resized to $112px$ sized square images and normalized. Throughout all experiments a batch size of 256 is used and models are trained for 20 epochs with early stopping. For all 4 student architectures experiments are performed

with the 3 different knowledge distillation techniques and training from scratch without any distillation. For vanilla knowledge distillation different values for the temperature are investigated. Conditional knowledge distillation is conducted with a temperature of 4. For the curriculum-based knowledge distillation linear and cosine decay are investigated. As performance metrics we report validation accuracy and validation $F1$-score. For size and speed metrics, the inference and train speed are reported as well as the model size in Megabytes and the number of trainable parameters. All experiments are conducted with a single *Nvidia A40* GPU.

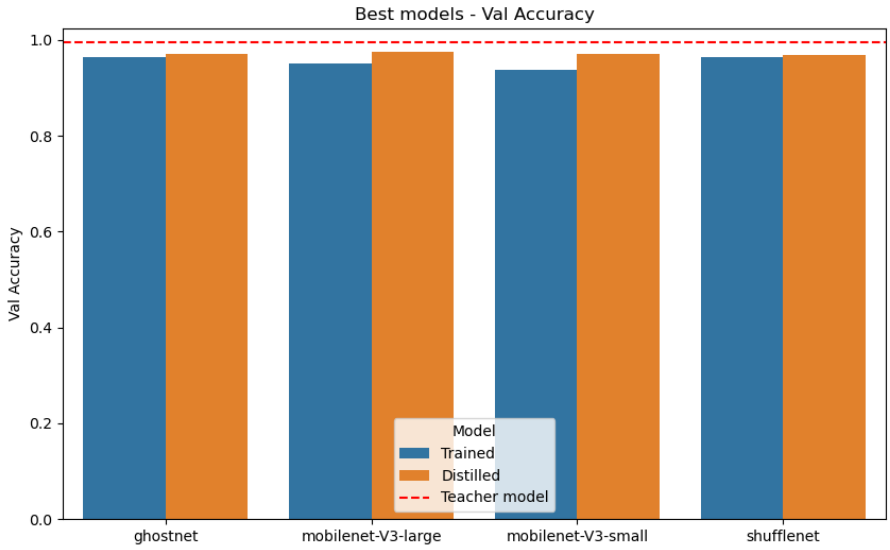

Figure 2: Validation Accuracy of Different Student Architectures

6 Results

Figure 2 depicts the validation accuracy of the teacher model, represented by a red dashed line, along with the best model trained with Knowledge Distillation (KD) and the model trained from scratch for each of the four student architectures. Notably, the validation accuracies of the models trained with KD closely align with that of the teacher model, indicating successful knowledge transfer. They also marginally yet consistently outperform the models trained from scratch for each student architecture. This indicates that all architectures perform comparably well and that KD offers a consistent advantage. The validation accuracy is very close to the teacher for all student architectures, as shown in Figures 3, 4, 5 and 6.

Our experiments revealed distinct optimal hyperparameters for each student architecture. The GhostNet model achieved its best performance using curriculum learning with a linear decay schedule for temperature and a loss rate of 0.5. For the MobileNet V3 Large and ShuffleNet models, optimal performance is obtained with a temperature parameter of 2 and

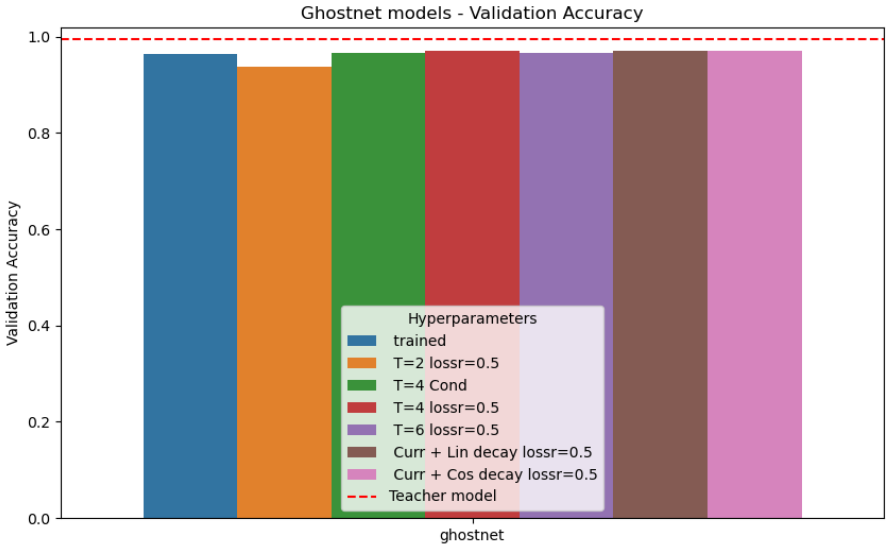

Figure 3: Validation Accuracy of GhostNet

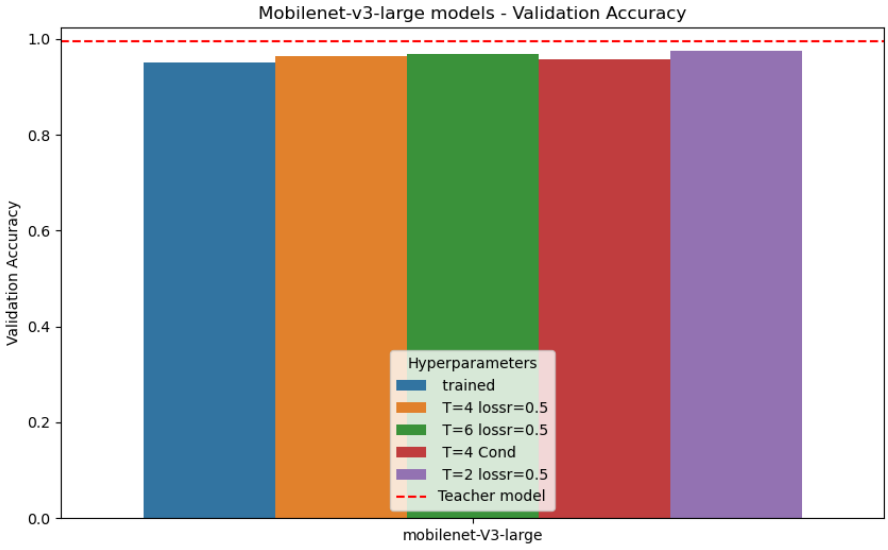

Figure 4: Validation Accuracy of MobileNet V3 Large

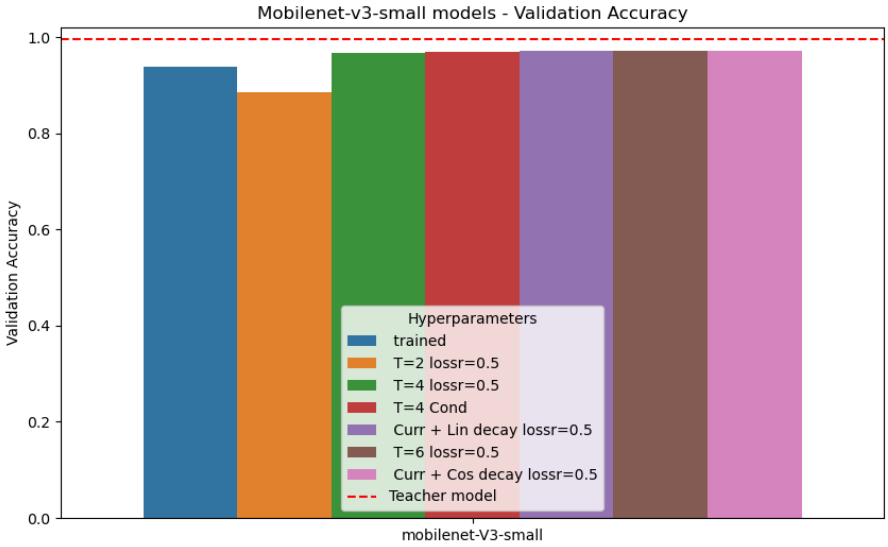

Figure 5: Validation Accuracy of MobileNet V3 Small

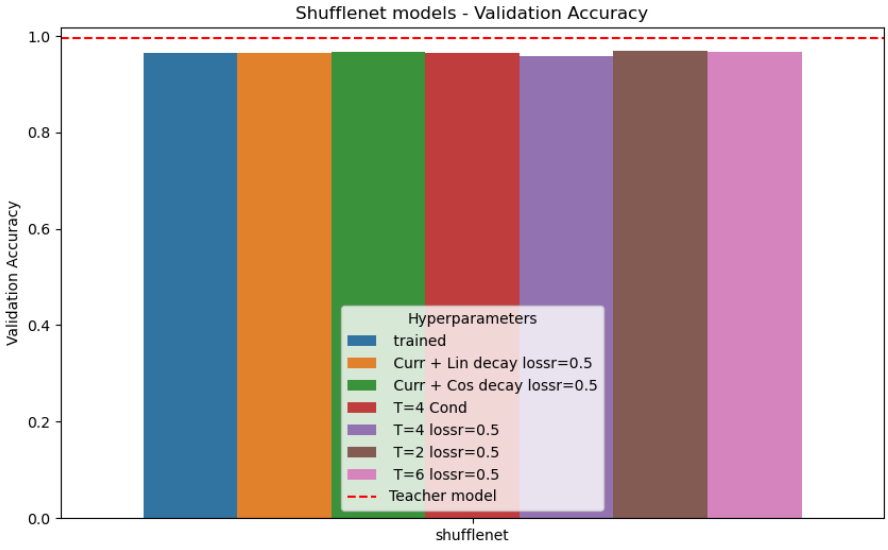

Figure 6: Validation Accuracy of ShuffleNet

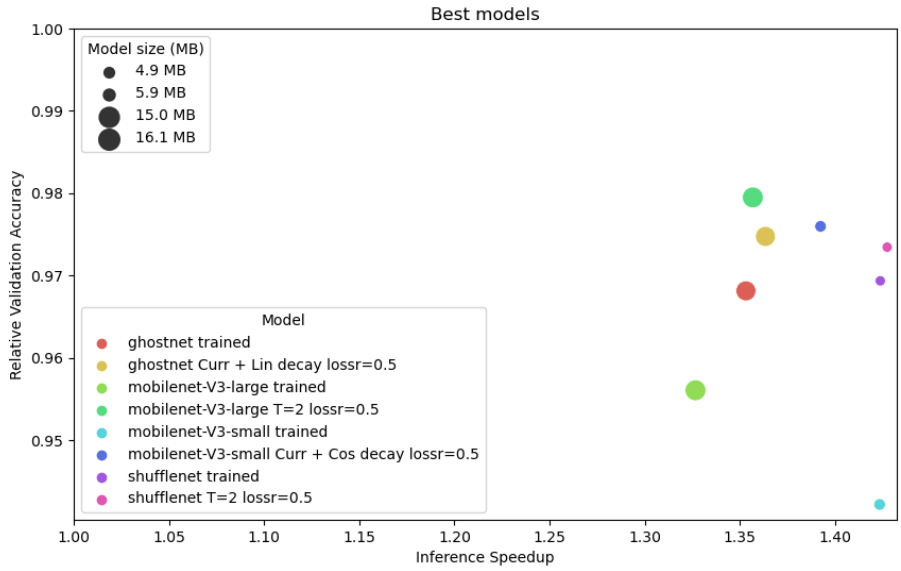

Figure 7: Relative Validation Accuracy and Inference Speedup of Different Student Architectures Trained with KD and from Scratch

a loss rate of 0.5. On the other hand, the MobileNet V3 Small model performs best when curriculum learning is combined with cosine decay schedule for temperature and a loss rate of 0.5.

Figure 7 depicts a scatter plot that illustrates the relative validation accuracy (compared to the teacher model) and inference speedup of each student model, both for models trained with KD and those trained from scratch. The thickness of each point represents the memory footprint of the model in MB.

All student models, when trained with KD, achieved similar performance, with relative validation accuracies ranging from 0.975 to 0.98 and inference speedups between 1.35 and 1.4 times. However, models trained from scratch showed a decrease in relative validation accuracy down to 0.95. Interestingly, there are some variations in inference speed even between models with the same architecture which could be attributed to slightly inconsistent allocation of hardware resources or variations in memory management during inference operations. Despite these variations, inference speed is generally close across all architectures.

Among all models, MobileNet V3 Large had the highest relative validation accuracy at 0.98 while MobileNet V3 Small and ShuffleNet achieved slightly higher inference speedups of 1.4 times.

These findings suggest that student models trained with KD not only match the performance of their teacher models but also benefit from faster inference speed and lower memory usage.

Figure 8 provides a focused view on only those models trained with KD, further emphasizing

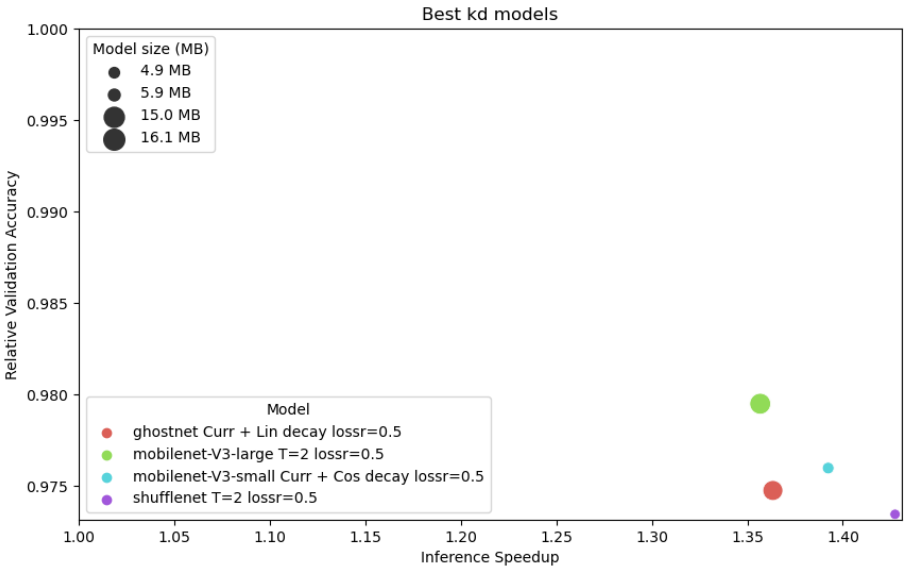

Figure 8: Relative Validation Accuracy and Inference Speedup of Different Student Architectures Trained with KD

these observations.

We also conducted the same experiments on a CPU and achieved significant inference speedups compared to those run on a GPU. Figure 9 illustrates inference speedups ranging from 5 to 9 times over the teacher model. ShuffleNet offers the highest inference speedup, slightly over 9 times, with a relative validation accuracy of 0.973. The large and small MobileNet architectures are roughly similar. Although the larger architecture delivers slightly better relative accuracy performance (0.979) compared to the smaller one (0.976), the latter appears to be slightly faster than the former. GhostNet yields the smallest inference speedup, at 5 times, while maintaining 0.975 of the teacher model's validation accuracy.

7 Limitations

Due to the limited available time for the project and project specific circumstances multiple limitations have to be considered regarding the achieved results. Because of significant training times and the time constraints set by the project course, the search space for different hyperparameter combinations was heavily restricted. With the same reasoning no statistical significance testing was performed and experiments for each setup only performed once. This weighs even more heavy for this project since the student models reach very good results even without knowledge distillation and the different distillation techniques are very close in performance. Therefore, the results can only be interpreted as indications, especially regarding the differences between different knowledge distillation techniques. For a few experiments a batch size of 32 was also tested and achieved slightly better results but due to the increased

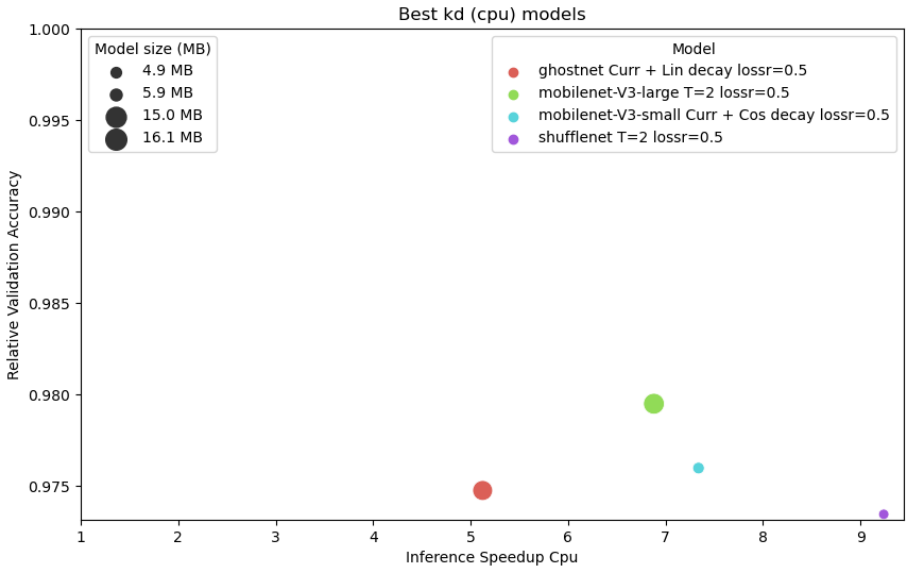

Figure 9: Relative Validation Accuracy and Inference Speedup of Different Student Architectures Trained with KD

training time with a reduced batch size, this was not further pursued. The captured data for training and inference speed is very noisy, likely because of the shared nature of the computing resource, therefore for speeds the median batch time or the minimum epoch times are reported.

8 Discussion

This study aimed to apply knowledge distillation (KD) techniques to echocardiogram view classification by comparing the performance of models trained with KD to those trained from scratch. The results indicate that KD offers a consistent advantage over models trained from scratch, regardless of the student architecture used. This is evident from the validation accuracies of models trained with KD, which closely align with that of the teacher model and outperform models trained from scratch.

The use of KD in this context has significant practical implications: by reducing memory requirements and increasing inference speed, these models could potentially be deployed locally in resource-constrained environments such as medical equipment. This would enable real-time prediction of echocardiogram views, leading to more efficient and accurate diagnoses in the medical field.

However, it's important to note that due to significant training times and project course time constraints, the search space for different hyperparameter combinations was heavily restricted. This limitation could potentially be addressed in future research by allowing for

a larger search space for hyperparameter combinations and conducting multiple runs of each experiment setup to ensure statistical significance.

Interestingly, by using a relatively small (16 layers) VGG16 model as the teacher model, the student architectures achieved only small increases in inference speed. This suggests that other compression techniques might be more suitable if the teacher model is already relatively small.

In conclusion, this study underscores the effectiveness of KD in improving the performance of student models in echocardiogram view classification. It also emphasizes the importance of selecting appropriate hyperparameters based on the specific student architecture used. Future research could explore more student architectures and other model compression techniques such as quantization and pruning or a combination thereof.

9 Conclusion

This study successfully demonstrates the application of knowledge distillation (KD) techniques in echocardiogram view classification. The conducted experiments reveal that KD offers a consistent, albeit small, advantage over models trained from scratch, regardless of the student architecture used. This is evident from the validation accuracies of models trained with KD. These accuracies closely align with that of the teacher model and marginally outperform those of models trained from scratch.

Additionally, the study highlights that different student architectures require distinct optimal hyperparameters for best performance. For instance, GhostNet achieved its best performance using curriculum learning with a linear decay schedule for temperature and a loss rate of 0.5, while ShuffleNet performed best with a temperature parameter of 2 and a loss rate of 0.5.

Furthermore, all student models trained with KD achieved similar performance, with relative validation accuracies ranging from 0.975 to 0.98 and inference speedups between 1.35 and 1.4 times. However, models trained from scratch showed a decrease in relative validation accuracy down to 0.95.

Among all models, MobileNet V3 Large had the highest relative validation accuracy at 0.98 while MobileNet V3 Small and ShuffleNet achieved slightly higher inference speedups of 1.4 times.

These findings confirm that student models trained with KD not only match the performance of their teacher model on this task but also benefit from faster inference speed and lower memory usage.

However, due to significant training times and project course time constraints, the search space for different hyperparameter combinations was heavily restricted. For the same reasons, no statistical significance testing was performed, and each experiment setup was only performed once. Therefore, these results can only be interpreted as indications, especially regarding inference speed differences between different KD techniques.

In conclusion, this study underscores the effectiveness of KD in improving the performance

of student models in echocardiogram view classification. It also emphasizes the importance of selecting appropriate hyperparameters based on the specific student architecture used.

Acknowledgment

"The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers e-Commons at Chalmers partially funded by the Swedish Research Council through grant agreement no. 2022-06725."

References

[1] Echonet-dynamicA Large New Cardiac Motion Video Data Resource for Medical Machine Learning.

[2] Echonet-lvhA Large Parasternal Long Axis Echocardiography Video Data Resource.

[3] Tufts medical echocardiogram dataset (tmed) A multi-task SSL benchmark for classifying view and diagnosing heart disease severity.

[4] Jimmy Ba and Rich Caruana. Do Deep Nets Really Need to be Deep? In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[5] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-Layer Distillation with Semantic Calibration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7028–7036, 2021.

[6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 742–751. Curran Associates Inc., 2017.

[7] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning Student Networks via Feature Embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):25–35, 2021.

[8] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[9] Jindong Gu and Volker Tresp. Search for Better Students to Learn Distilled Knowledge. *ECAI 2020: 24th European Conference on Artificial Intelligence*, pages 1159–1165, 2020.

[10] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge Distillation of Large Language Models, 2023. 10.48550/arXiv.2306.08543.

[11] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network, 2015. 10.48550/arXiv.1503.02531.

[13] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for

mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[14] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017. 10.48550/arXiv.1704.04861.

[15] Jangho Kim, Minsung Hyun, Inseop Chung, and Nojun Kwak. Feature Fusion for Online Mutual Knowledge Distillation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4619–4625, 2021.

[16] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised Knowledge Distillation Using Singular Value Decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–350, 2018.

[17] Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few Sample Knowledge Distillation for Efficient Network Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14639–14647, 2020.

[18] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1504–1512, 2023.

[19] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional teacher-student learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6445–6449. IEEE, 2019.

[20] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved Knowledge Distillation via Teacher Assistant. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5191–5198, 2020.

[21] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[22] Nikolaos Passalis and Anastasios Tefas. Learning Deep Representations with Probabilistic Knowledge Transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[23] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous Knowledge Distillation Using Information Flow Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2020.

[24] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. ALP-KD: Attention-Based Layer Projection for Knowledge Distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13657–13665, 2021.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch:

An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[26] Mary Phuong and Christoph Lampert. Distillation-Based Training for Multi-Exit Architectures. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1355–1364, 2019.

[27] Sonia Shah, Matthew W Segar, Nitin Kondamudi, Colby Ayers, Alvin Chandra, Susan Matulevicius, Kartik Agusala, Ron Peshock, Suhny Abbara, Erin D Michos, et al. Supranormal left ventricular ejection fraction, stroke volume, and cardiovascular risk: findings from population-based cohort studies. *Heart Failure*, 10(8):583–594, 2022.

[28] Sungho Shin, Yoonho Boo, and Wonyong Sung. Knowledge distillation for optimization of quantized deep neural networks. In *2020 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 1–6, 2020.

[29] Hui Wang, Hanbin Zhao, Xi Li, and Xu Tan. Progressive blockwise knowledge distillation for neural network acceleration. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, pages 2769–2775. AAAI Press, 2018.

[30] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L. Yuille. Snapshot Distillation: Teacher-Student Optimization in One Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

[31] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2019.

[32] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019.

[33] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[34] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.