Knowledge Distillation

Deep Mutual Learning

Team Purple:

Mamtha Mahanthesh Vathar (124106172)
Ramya Thimmappa Gowda (124106243)
Moras Kashyap (124103079)
Akash Vijaykumar Kotekar (124106044)
Harsh Gaur (124103720)
Mandar Deepak Bagwe (123111875)
Meghan Vinayak Mane (124100669)

GitHub: https://github.com/Akeshko/Knowledge-Distillation

INDEX

MOTIVATION

With the rapid advancement in Deep Learning technology, deep learning models these days have become very powerful. But at the same time, they require a huge amount of computing power and memory. Therefore, with an increase in its demand in the day-to-day world, there is a need for a model that should be smaller, compact, faster but still quite accurate. Knowledge Distillation tackles this problem by providing an effective solution. However, traditional Knowledge Distillation usually follows a unidirectional learning process. Therefore, we also implemented deep mutual learning, where the student models learn collaboratively from each other, and thus can make this process of knowledge transfer even more efficient.

INTRODUCTION

Knowledge Distillation is a technique through which the process of transfer of knowledge happens from a larger (teacher) model to a smaller (student) model. The main goal of this is that, once it is completed, the student model, even though it is much smaller in size and need much less computational power, is able to perform in the similar way as the teacher model is performing.
This process is similar to what happens in a class. Like in a class, the teacher shares its information with the students present in the class, similarly, in knowledge distillation, a large teacher model shares its knowledge with a smaller student model. By focusing on the outputs provided by the teacher, the student model learns to mimic the behaviour of the teacher model, so that it can make similar guesses as the teacher model.

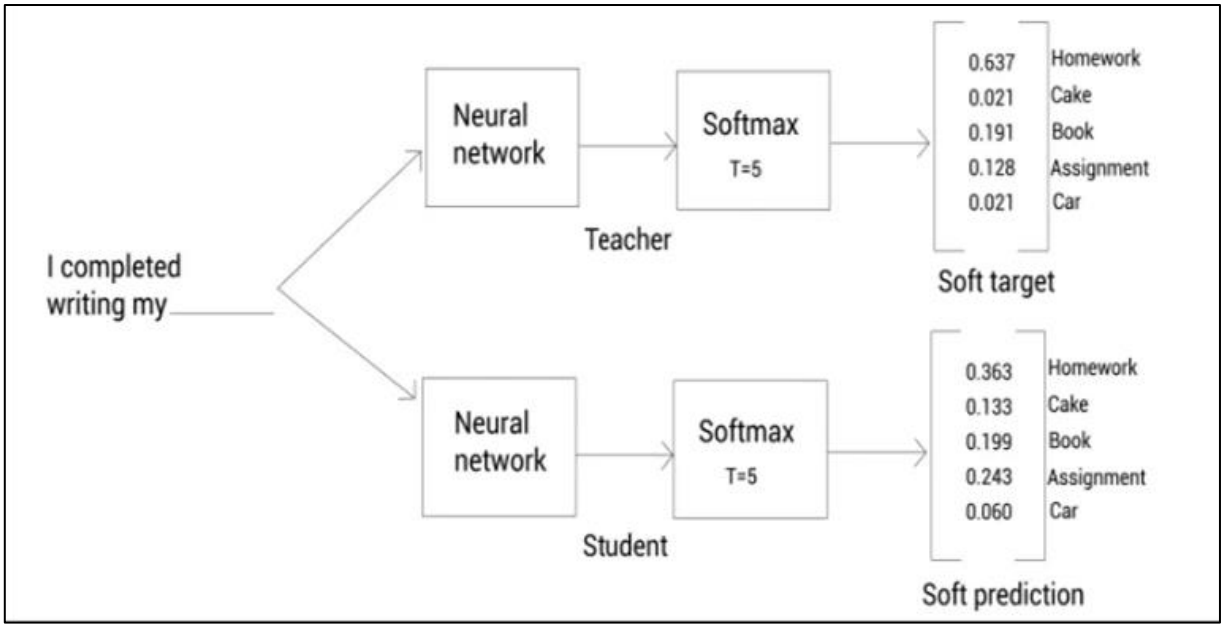


Figure 1: Knowledge Distillation

In the image diagram above, there is a teacher model as well as a student model. To predict the next word in a phrase, both of the models are using a softmax function that has an elevated temperature (T= 5). The distribution of the teacher is called soft target. For different possible words, like homework, cake, book etc, soft target shows the probability. These smooth and higher temperature outputs are then used by the student to learn not only the final choice but also the near miss options rankings made by the teacher. This extra information is called dark knowledge. It is very useful in helping the student model to mimic the nature of the teacher model.

PROJECT TIMELINE

We implemented this project in a three weeks window period.

WEEK 10 – PLANNING AND DESIGNING

In week 10, we started off with clearly outlining the goals of the project- compress and optimize a large teacher model using knowledge distillation. After this we selected adequate models. We went for ResNet-50 for the teacher, ResNet-18 for the student. Then we chose the algorithms we would be implementing. (Soft Targets, Attention Transfer, and Deep Mutual Learning). In tandem, we selected 3 datasets- CelebA, CIFAR-10, MNIST and then we finalized pinned down evaluation metrics such as accuracy, inference speed, and model size.

WEEK 11 – IMPLEMENTATION

In this week, on each dataset, first we trained the teacher model until it achieved high accuracy. Then we trained initial student models. Then we trained the student models using Soft Target Distillation and attention transfer. As a result of this, we got baseline results on accuracy, speed, and parameter counts. For subsequent improvements, we also documented any gaps between the teacher and the student models.

WEEK 12 – IMPROVEMENT

In the final week, we did hyper-parameter tuning like adjusting the temperature for soft targets, for distillation loss adjusting weighting factors and so on. We then implemented Deep Mutual Learning and observed how the convergences are affected by this sharing of the knowledge. Then finally we compared the student models.

METHODOLOGY

DATASETS

We selected three diverse datasets to do efficient evaluation of our approach. These datasets range from simple and straightforward digit classification database to challenging facial attribute tasks containing database. This selection ensures that we are applying our approach over varying degrees of difficulty.

CelebA

This is a very large-scale dataset that has facial attributes of over 200,000 images. Each image is annotated with 40 binary labels such as eyeglasses, male etc. This dataset is very suitable to help the model to learn subtle variations in appearance.

Figure 2: CelebA dataset

CIFAR-10

This dataset has 60,0000 images which are evenly distributed across 10 different classes. This dataset is of moderate complexity. In our experiments it helped us to know how the models are able to handle different categories of the object.

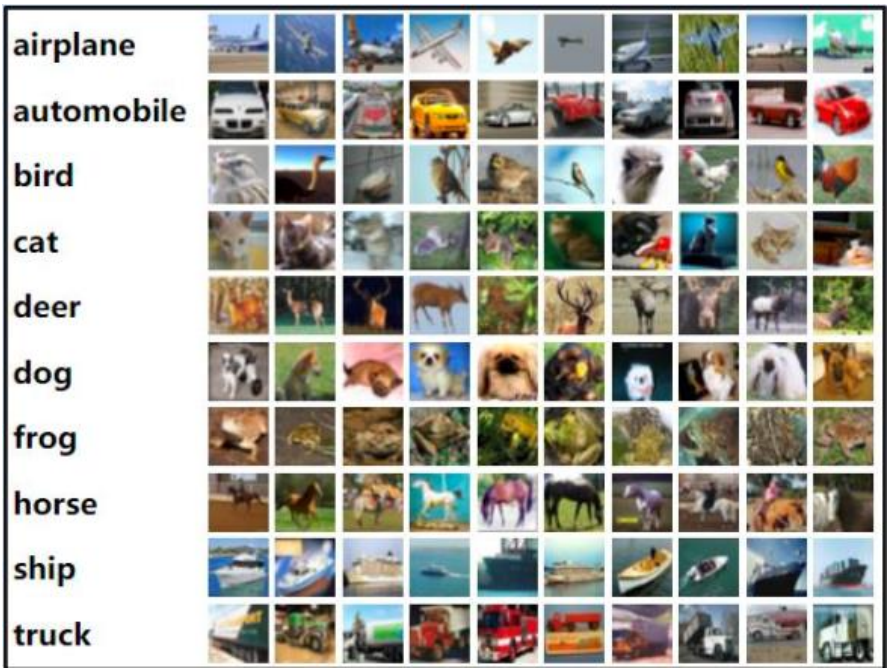


Figure 3: CIFAR-10 dataset

MNIST

This is a standard dataset on handwritten digital images. In terms of complexity level, it comes under the category of being simple, but it is a quite reliable benchmark for the classification tasks. It lets us know how good is our model is in terms of learning from minimal visual complexity.

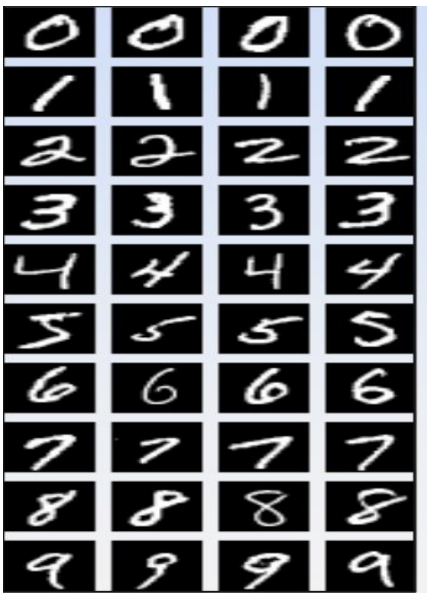


Figure 4: MNIST dataset

MODELS

Teacher Model: ResNet-50

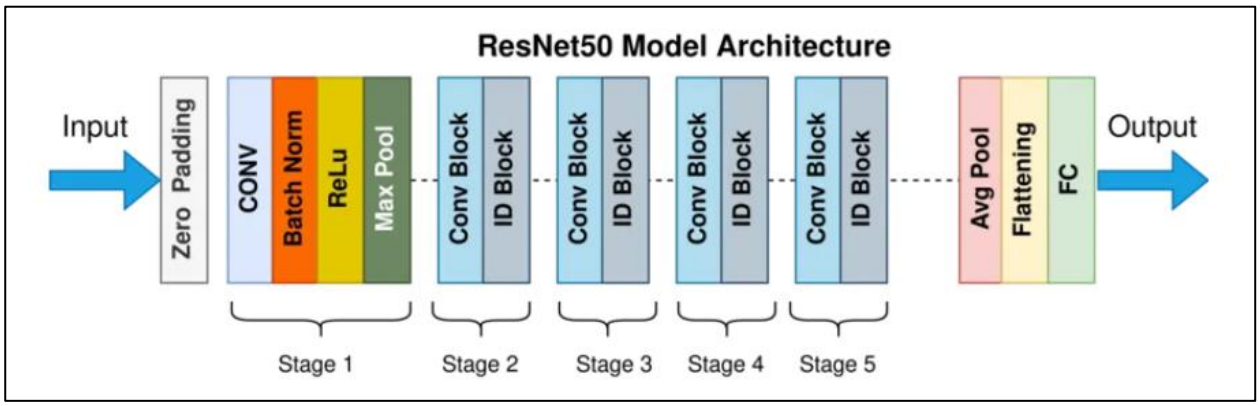


Figure 5: ResNet50 Architecture

We decided to choose ResNet-50 as the teacher model. It has decent depth which is around 50 layers and it have very good ability to learn representations that are rich and hierarchical. Therefore, it makes a very good teacher model that can be used for distilling knowledge into a smaller student.

Student Model: ResNet-18

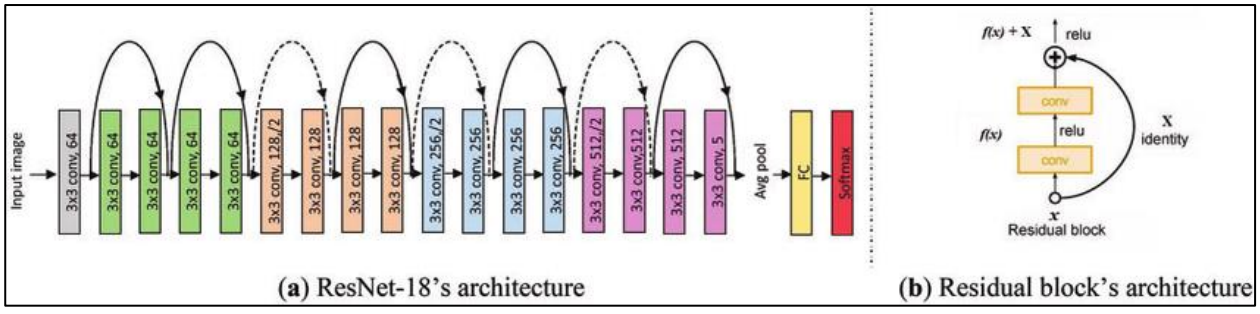


Figure 6: ResNet18 Architecture

ResNet-18 has a shallower architecture, it has around 18 layers and far less parameters. However, still it's quite efficient for quick to train and deploy.

By choosing these two separate models, we made sure to keep a balance between high accuracy and less computational overhead.

ALGORITHMS

Soft Target Distillation

In this form of distillation technique, the student model uses the soft outputs made by temperature-scaled softmax of the teacher, in the learning process. This information helps the student to learn the inter-class relationships and thus as a result of which the generalisation of the student model is improved.

Attention Transfer

In this form of distillation technique, attention maps of the teacher are transferred to the student, so that it can focus and learn the most crucial regions of an image. Here, the student aligns its intermediate feature activations with that of the teacher and as a result of which it focuses only on key visual cues and not unnecessary details or noise.

Deep Mutual Learning

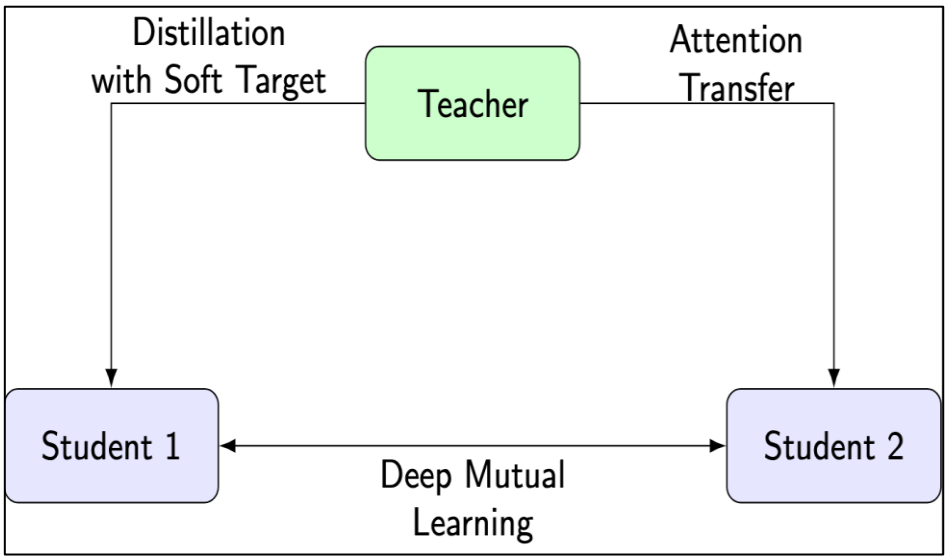


Figure 7: Deep Mutual Learning

In Deep Mutual Learning, the student models refine the prediction of each other by the means of mutual feedback with its peer. The result of this is that student models are able to learn more a more stable and efficient manner.

Overall, this methodology establishes a pipeline, that has a teacher model with quite high capacity, which transfers its final logits (in form of soft targets) and feature – level insights (in form of attention maps) to the student model, which then learns how to mimic the output of the teacher model. After this in the pipeline, DML is applied so that the student models can refine each other outputs and thus try increasing the performance of the student models.

TECHNICAL OVERVIEW

The architecture shown in Figure 8 illustrates a knowledge distillation setup on a dataset with deep neural networks and different training strategies. The main idea is to transfer knowledge from a larger, more complex teacher model (ResNet-50) to smaller student models (ResNet-18). At the same time, the approach encourages mutual learning between different distillation techniques, helping the student models learn more effectively and boost their performance.

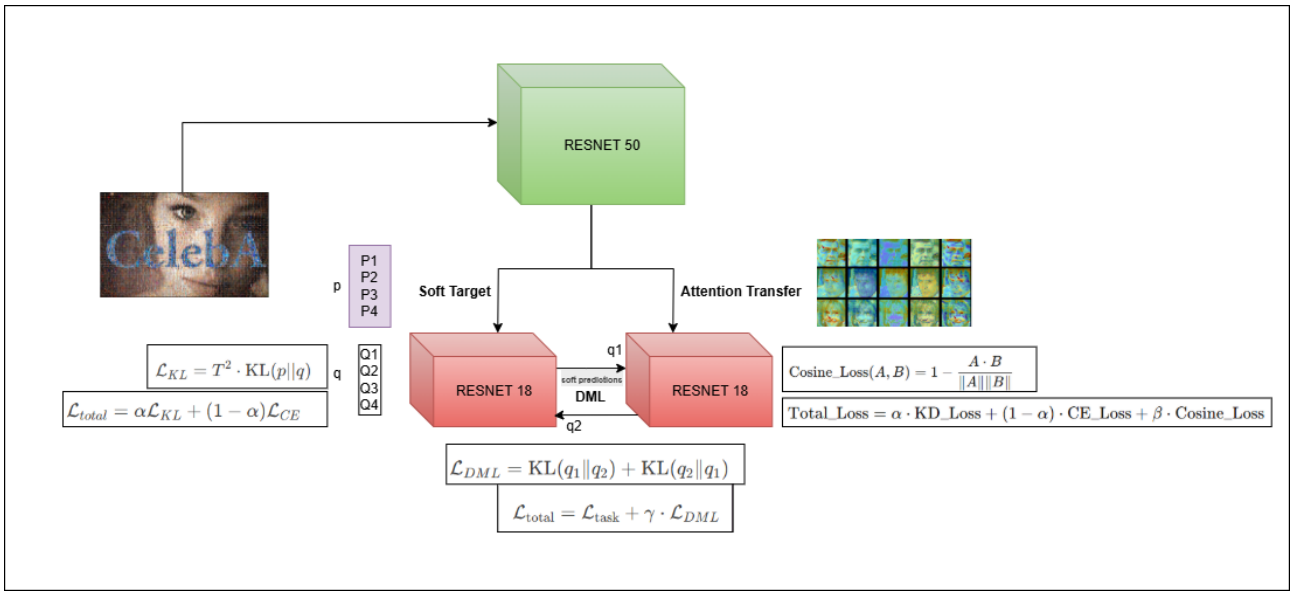


Figure 8: Architectural Overview

TEACHER MODEL (ResNet-50)

The method begins with a dataset, such as CelebA, being fed into a pre-trained ResNet-50 model, which serves as the teacher. This model provides two significant outputs:

i. Soft Targets:

Instead of giving just one correct class label, the teacher provides a list of class probabilities. This produces a more detailed learning signal by improving the student models in understanding the relationships between various classes.

ii. Attention maps:

These are illustrations of the areas of a picture that the instructor model concentrates on during prediction. By highlighting important elements in the picture, they help the student models.

STUDENT MODEL (ResNet-18)

Two ResNet-18 models were used as student networks. Each student was trained from scratch with a two-phase process:

i. Stage 1 – Traditional KD: The student models mimicked the teacher using the teacher models soft targets and attention maps.

1. Soft target knowledge distillation:

Here the student is trained to align its output with the soft targets provided by the teacher. This is accomplished using the Kullback-Leibler (KL) divergence, which is scaled by temperature T.

2. Attention Transfer:

Feature maps from the teacher's selected intermediate layers, Layers 2 and 4, were utilized to generate attention maps, which were then used to train the student model. Layer 2 was chosen because it captures moderately abstracted feature maps that blend low-level and mid-level visual features. These features help the student learn generalized representations early in the network. Layer 4 was chosen because it captures all the high-level semantic features essential for the final classification. This helps the student model understand how the teacher makes its decisions, not just what the final answers are.

ii. Stage 2— Deep mutual learning: In addition to learning from the teacher, the student models were encouraged to share their knowledge. Each student treats the other's predictions as a soft target, reducing bidirectional KL divergence.

HARDWARE AND SOFTWARE

All the models were implemented using the PyTorch deep learning framework, leveraging its modular design and GPU support.

Training Environment:

a. Google Colab Pro (with Tesla T4/A100 GPUs) for rapid prototyping and experimentation.

b. Local GPU machines (Nvidia RTX-series) for batch training and hyperparameter tuning

LOSS FUNCTIONS

1) Kullback-Leibler Divergence Loss (KL Divergence)

KL Divergence Loss measures divergence of probability distribution from one value to another. It is mostly used for Probabilistic Models.

$$\mathcal{L}_{KL} = \sum_{i=1}^{n} \sum_{j=1}^{k} y_{ij} \log\left(\frac{y_{ij}}{\hat{y}_{ij}}\right)$$

where,

n = number of data points

k = number of classes

y_{ij} = binary indicator (0 or 1) if class label j is the correct classification for data point i

$\hat{y}_{ij}$ = predicted probability for class j

2) Binary Cross-Entropy Loss (Log Loss)

Binary Cross-Entropy Loss, also known as Log Loss, is used for Binary Classification problems like CelebA classification where it measures performance of classification model whose probability values are between 0 and 1.

$$\mathcal{L}_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where,
n = number of data points
y_i = actual binary label (0 or 1)
$\hat{y}_i$ = predicted probability

3) Categorical Cross-Entropy Loss

Categorical Cross-Entropy Loss is used for Multiclass Classification problems like CIFAR-10 and MNIST classification where it measures performance of classification model whose probability distribution over multiple classes.

$$\mathcal{L}_{CCE} = -\sum_{i=1}^{n}\sum_{j=1}^{k} y_{ij}\log(\hat{y}_{ij})$$

where,
n = number of data points
k = number of classes
y_{ij} = binary indicator (0 or 1) if class label j is the correct classification for data point i
$\hat{y}_{ij}$ = predicted probability for class j

4) Cosine Proximity Loss

Cosine Proximity Loss measures cosine similarity between predicted and target vectors encouraging them to point in the same direction.

$$\mathcal{L}_{cs} = -\frac{1}{n}\sum_{i=1}^{n}\frac{y_i \cdot \hat{y}_i}{\|y_i\|\|\hat{y}_i\|}$$

where,
n = number of data points
y_i = actual value
$\hat{y}_i$ = predicted value

In MNIST and CIFAR-10,

a. For Soft Target, we used

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{KL} + (1 - \alpha)\mathcal{L}_{CCE}$$

b. For Attention Transfer, we used

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{KL} + (1 - \alpha)\mathcal{L}_{CCE} + \beta\mathcal{L}_{CS}$$

In CelebA,

a. For Soft Target, we used

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{KL} + (1-\alpha)\mathcal{L}_{BCE}$$

b. For Attention Transfer, we used

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{KL} + (1-\alpha)\mathcal{L}_{BCE} + \beta\mathcal{L}_{CS}$$

While performing DML, we calculated the Loss as,

$$\mathcal{L}_{DML} = \mathcal{L}_{KL}(q1 \,\|\, q2) + \mathcal{L}_{KL}(q2 \,\|\, q1)$$
$$\mathcal{L}_{total} = \mathcal{L}_{Task} + \gamma \cdot \mathcal{L}_{DML}$$

CHALLENGES AND SOLUTION

During the implementation of the project, several technical hurdles were encountered.

1. High Computational Cost of Training Deep Models

Challenge:

To training deep architectures like ResNet-50 from complete scratch was very computationally demanding. As a result, there was very long training periods which required too high GPU memory was encountered.

Solution:

By using transfer learning, set up the training utilizing pre-trained models as teacher networks. By doing so, faster convergence, improved generalization, and greatly shortened training time and resources were therefore possible.

2. Overfitting in Student Models

Challenge:

Due to limited capacity, student models were more prone to overfitting.

Solution:

Using Dropout, Weight Decay, and Early Stopping among other regularizing methods, we aimed to improve generality. This is helpful because this method reduces overfitting without compromising model correctness.

3. Instability in Deep Mutual Learning

Challenge:

During simultaneous learning that happens in DML, student models can cause each other to diverge during early training phases due to conflicting gradients.

Solution:

i. Stabilized parameter adjustments by means of a lower learning rate.
ii. Applying a Gradual Distillation Strategy with such that:
    - Models first learnt on their own under normal direction.

- Only after the first convergence was KL divergence loss gradually raised.

4. Hyperparameter Sensitivity

Challenge:

Model performance is highly sensitive to the distillation hyperparameters, particularly Temperature (T) and Alpha (α).

Solution:

a. Conducted a systematic grid search across Temperature and Alpha values.
b. We selected only the best combinations determined by training stability and validation accuracy. This guaranteed strong performance.

RESULT AND COMPARISON

In this section, the outcomes of Knowledge Distillation as well as Deep Mutual Learning experiments are going to be presented. The primary focus will be on these four matrices-

a. Accuracy
b. Model Compression
c. Inference Speed
d. Loss Trends

By evaluating our experiments on these key aspects, our aim is to demonstrate how the performance of a smaller student model (ResNet18) is influenced when it is learning from a larger teacher model (ResNet50), when different distillation techniques such as Soft Target Distillation, Attention Transfer and Deep Mutual Learning are applied.

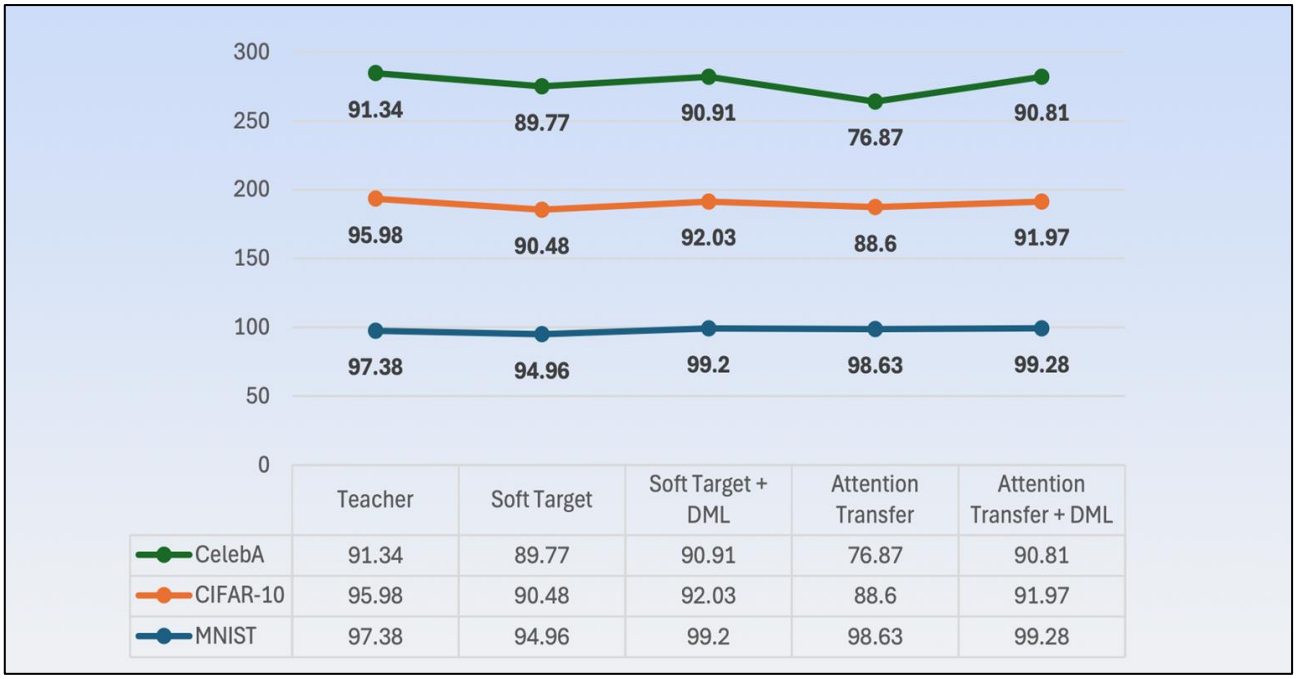
First, the overall accuracy achieved by each approach across three datasets- MNIST, CIFAR10, and CelebA will be studied. Following that, to demonstrate how our approaches capture complex class properties, we will highlight top-k accuracy measurements on CelebA. Next, the results of model compression will be described, demonstrating that model size can be significantly reduced without sacrificing performance. Later, we will discuss inference speed improvements, focusing on how this benefits real-world implementation. Finally, we will examine loss behaviours such as KL Divergence, Cross-Entropy, and so on in order to determine how well each method aligns the student's predictions with the teacher's outputs and ground truths.

ACCURACY

The accuracy metrics lets us know how close the predictions of the student model are to the ground-truth labels. Our experiments shows that Deep Mutual Learning consistently improves the student’s performance.

Specifically:

a. MNIST – We observed that using DML, the student model can match the accuracy level of the teacher or even exceed it. This is because the MNIST has relatively low complexity, as a result of which with the help of teacher’s guidance and mutual training with other students, the student is able to learn core features more easily.



	Teacher	Soft Target	Soft Target + DML	Attention Transfer	Attention Transfer + DML
CelebA	91.34	89.77	90.91	76.87	90.81
CIFAR-10	95.98	90.48	92.03	88.6	91.97
MNIST	97.38	94.96	99.2	98.63	99.28

Figure 9: Accuracy

b. CIFAR-10 - Here, the performance of the student, under Attention Transfer + DML and Soft Target + DML has noticeable gains when compared with the baseline training. However, it was still slightly below the teacher. CIFAR-10 is more complicated that MNIST, as a result of which the student is able to derive more benefit from mutual knowledge transfer, allowing it to focus on class differences that would be otherwise be missed.

c. CelebA – Unlike the other two datasets, CelebA uses fine-grained face attributes and multiple label output (40 binary attribute labels). This makes it quite difficult. Soft Target + DML outperforms single-directional distillation approaches, such as soft target distillation. Similarly, if the spatial attention maps are too granular, attention transfer may struggle, as evidenced by accuracy. Having said that, the combination of Attention-Transfer and DML enhances the attention transfer model quite a bit. This could be due to the fact that DML boosted the attention model by forcing it to focus on working on its soft probabilities as well.

TOP-K ACCURACY ON CELEBA

We measured Top-k accuracy to learn more about CelebA's performance. It assesses how well the model ranks the correct labels among its top predictions. Because CelebA contains numerous face features with little differences, top-k metrics indicate if the student model consistently predicts the correct attribute in its highest confidence predictions.

TOP-5 ACCURACY

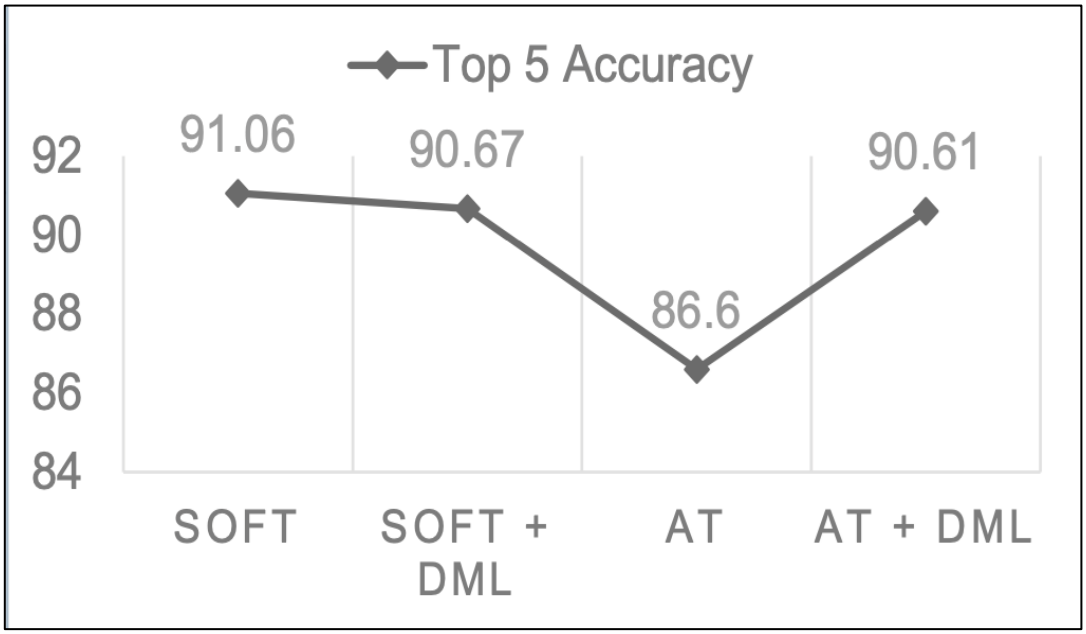


Figure 10: Top 5-Accuracy

a. Soft Targets + DML and Attention Transfer + DML has given similar results, which is around 90%-91%. This shows that with the help of teacher's output distribution and the feedback of the other peer, the student model is able to identify the correct facial attribute within its top 5 guesses most of the time.

b. Attention Transfer (AT) shows a slight drop. This might be because of the fact that when focused heavily on spatial features, broader and less localised cues in facial classification tasks can be missed.

TOP-10 ACCURACY

a. Methods combining DML maintain scores between 73% to 74%. This shows that even with more lenient criteria, attribute recognition is robust. Similar drop is observed as before for top 10 accuracy for attention transfer and we will continue to maintain the same reason.

b. With DML, soft targets still are able to rank decently, though not as good as Top-5 accuracy. We need to examine if it will be similar for Top 20 accuracy.

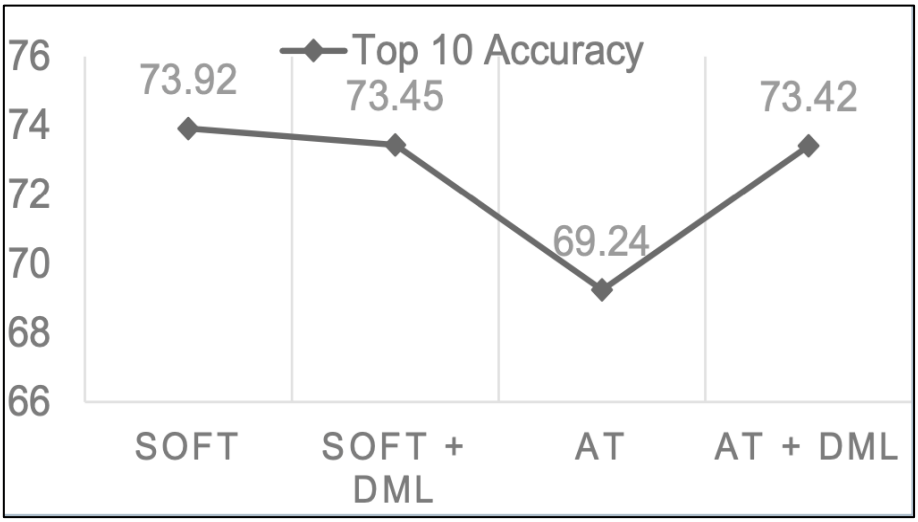


Figure 11: Top 10-Accuracy

TOP-20 ACCURACY

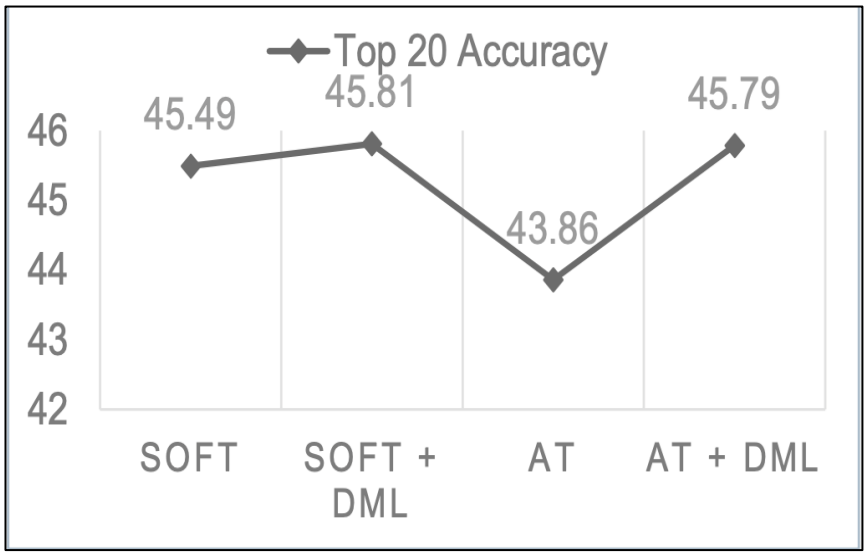


Figure 12: Top 20-Accuracy

All the methods are in the mid 40% range, with soft target + DML portraying a small edge. Here differences are less pronounced because of the presence of the wider threshold. Having said that, still the collaborative learning in DML gives slight gains.

Note: In classification tasks we generally observe that Top 10 accuracy and Top 20 will always be greater than Top5, but such is not the case for Celeb A.

The decrease observed in Top-10 and Top-20 accuracy for the CelebA dataset when compared to Top-5 is likely due to thresholding and the multi-label nature. Unlike single-label classification, CelebA predicts many attributes for each image. When thresholding is used to turn model outputs into binary decisions, it can sometimes suppress valid predictions, especially if we include more low-confidence in higher Top-K evaluations. This results in inaccurate label inclusion and a possible drop in accuracy beyond Top-5.

MODEL COMPRESSION

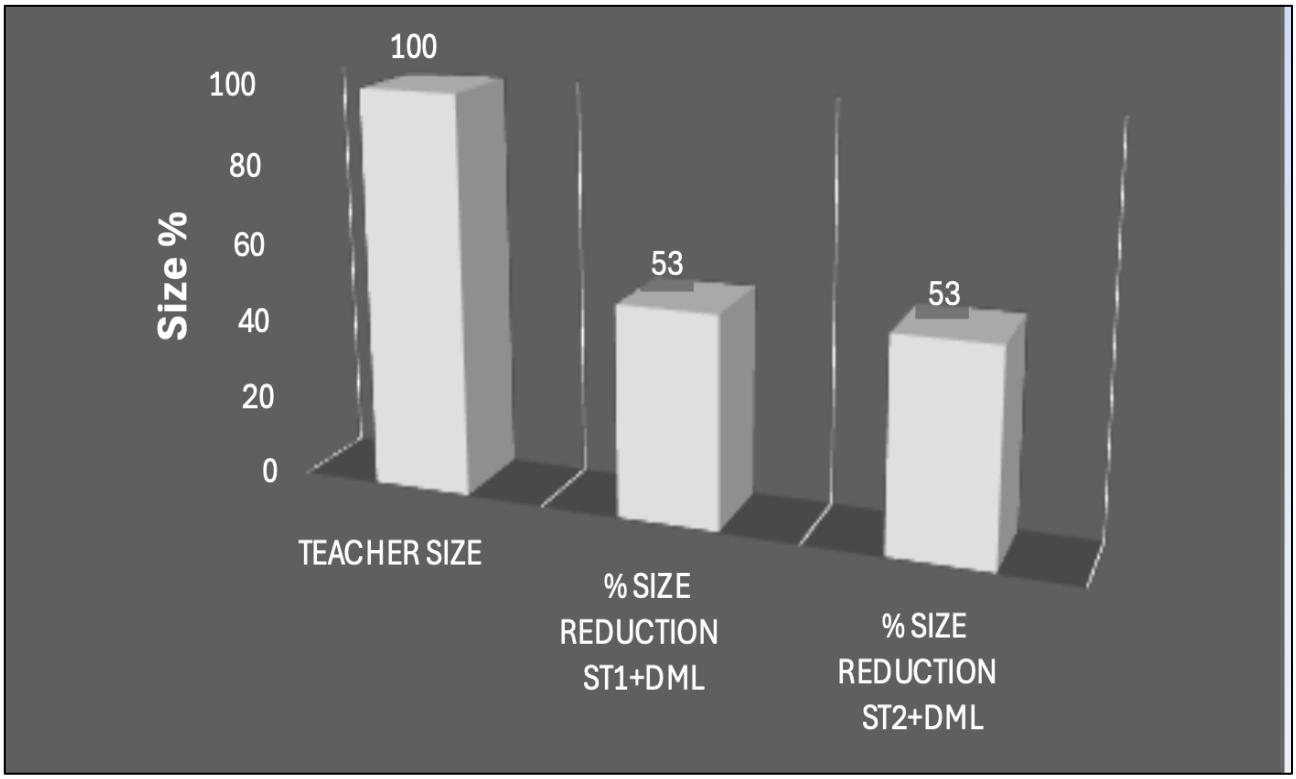


Figure 13: Model Comparison between MNIST, CIFAR10 and CelebA

The bar chart above compares the model sizes. Here, it can be seen that the ResNet-50 teacher model has been given a 100% baseline size. The two student models, one being Soft Target + DML and other being Attention Transfer + DML achieved 53% reduction in size of the teacher. This shows that student networks have significant reduction in its parameters. Therefore, it can be concluded that student models are way more memory efficient as well as easier to deploy them on small devices that are resource-constrained.

In this setup, the trade-off between the size of model and the performances is very less. The addition of DML ensures that the student models produce competitive results when compared with the teacher model, despite the fact that the smaller model occasionally exhibits a slight loss in accuracy when its outputs are compared with those of the teacher model. We were able to condense the neural network while ensuring that the accuracy penalty was as small as possible by employing DML training to improve the student model's predictions. The balance between the reduced processing footprint and the retained precision that we identified is crucial for real-world practical applications where speed and accuracy are both crucial.

INFERENCE SPEED

In real-time and large-scale applications, inference speed is a key factor. Through our experiments, we show that the combination of model compression and knowledge distillation significantly boosts efficiency. The following results shows the improvements for MNIST, CelebA and CIFAR-10.

a. MNIST-

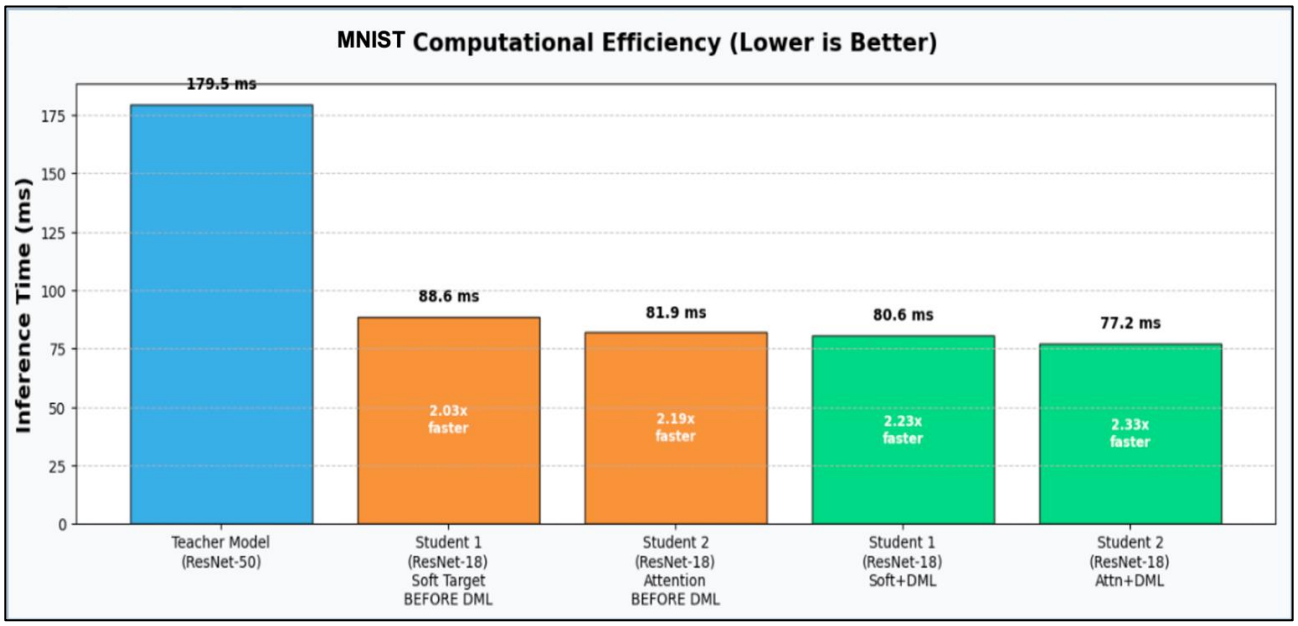


Figure 14: MNIST Computational Efficiency

i. The teacher model has deeper architecture when compared to the student's model. Therefore, even though it has high accuracy, it requires a good amount of computational time.

ii. Additionally, the student model, which is distilled through either Soft Target or Attention Transfer, operates at a speed that is more than double that of the original model.

iii. Following DML's optimisation, an extra 1% to 2% speedup is observed.

b. CelebA -

i. Due to the growing need for more intricate feature extraction, there is a significant level of inference in this case. The existence of fine-grained facial features is the cause of this rise in demand.

ii. Up to 3.3 times faster results are obtained when the bigger ResNET-50 is replaced with a smaller yet DML-enhanced ResNet-18. This increase can be attributed to two factors: first, the compressed student model has a lot less parameters, and second, DML greatly improves the learning signals.

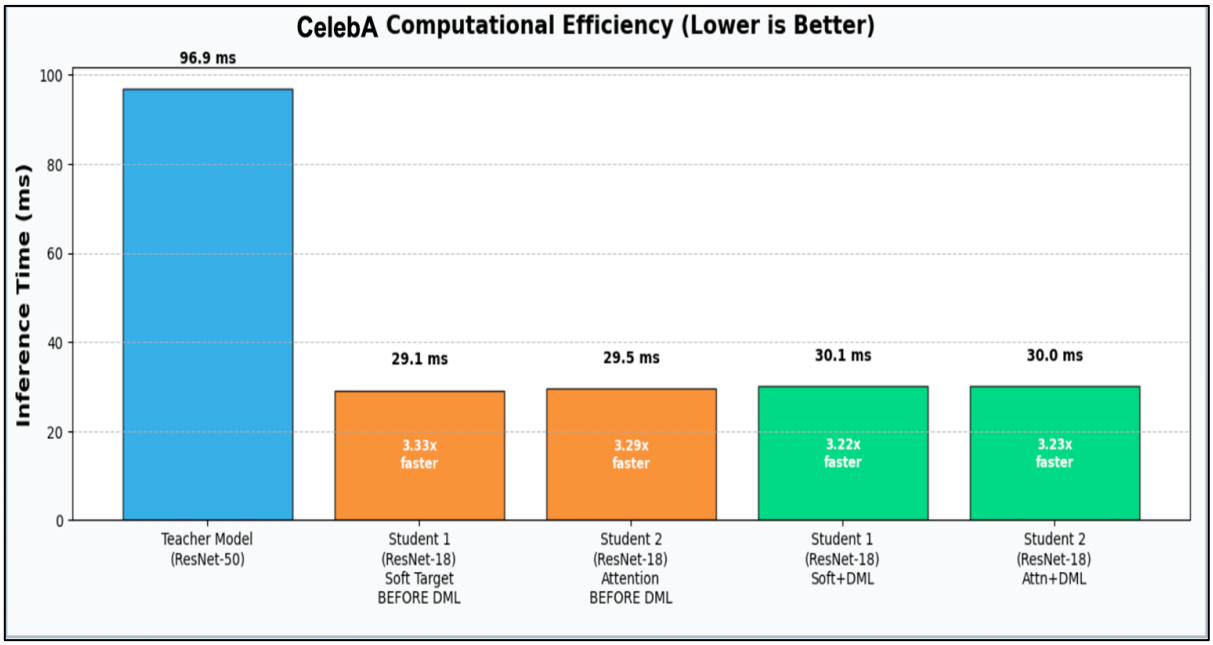


Figure 15: CelebA Computational Efficiency

c. CIFAR-10 -

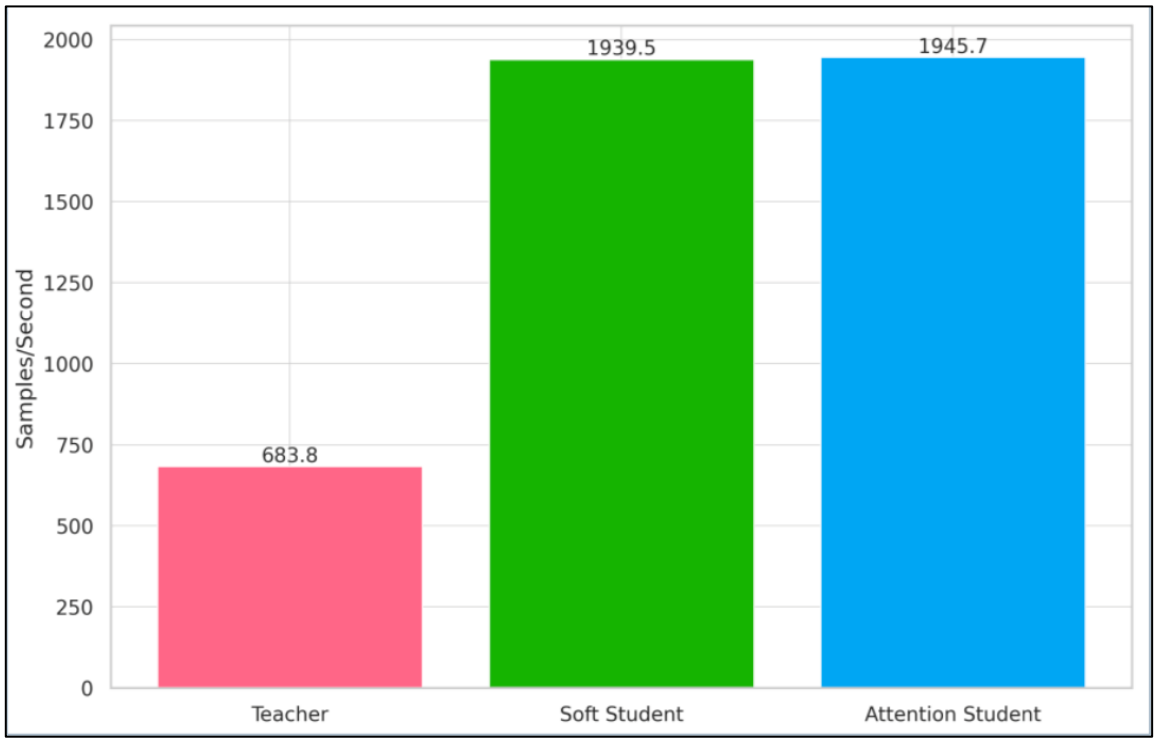


Figure 16: Comparison of Inference speed in terms of Samples per seconds.

i. For CIFAR10, the inference speed is three times faster for students.

ii. The teacher model, on the one hand, processes about 684 images every second. Conversely, soft student and attention student process about 1940 images each second.

iii. These results shows that the faster forward passes are achieved due to reduced depth and parameters. Moreover, the impact on the accuracy remains minimal.

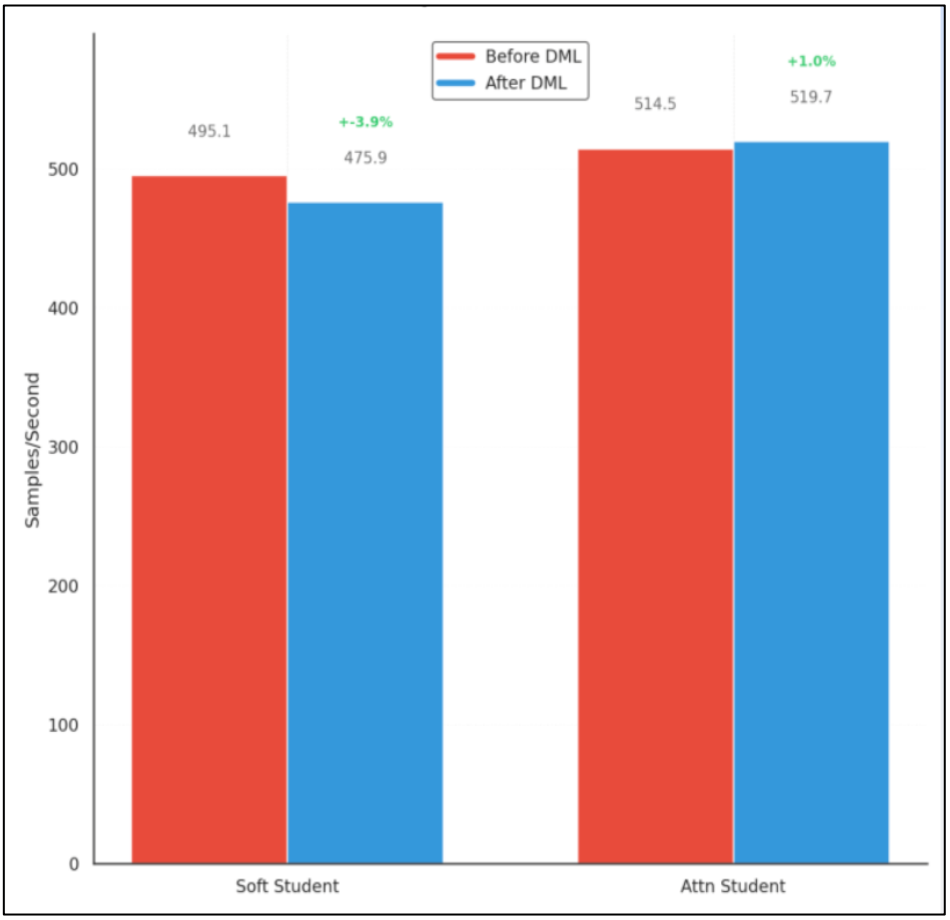


Figure 17: Inference Speed: Before the application of DML vs After its application

iv. The above visual illustrates how the students can cover more stable and efficient internal representations with the aid of DML. Consequently, following DML, the throughput is increased by an additional 1 to 4 percent.

In summary, light student architectures, particularly those to which DML is applied, have consistently produced faster inference speeds across all investigated datasets.

Note: Compared to CelebA, higher inference time was seen for simpler datasets like MNIST. Due to its ability to better utilise the GPU's parallel capabilities, this shows that the T4 GPU environment affects the inference timings differently due to the nature of data and resource utilisation capabilities.

FEATURE MAPS

Using the visualizations of feature map, it can be easily visualized and thus seen how the images are being processed internally by each network. In our CIFAR-10 experiments, in the regions corresponding to key focused contours, the teacher is producing more defined activations. On the other hand, the student shows broader or less focused patterns occasionally. Figure 18 is attention map comparison for lower layer (layer 2). This distinction is most apparent here. The image above shows that the teacher model is better at capturing delicate textures and edges.

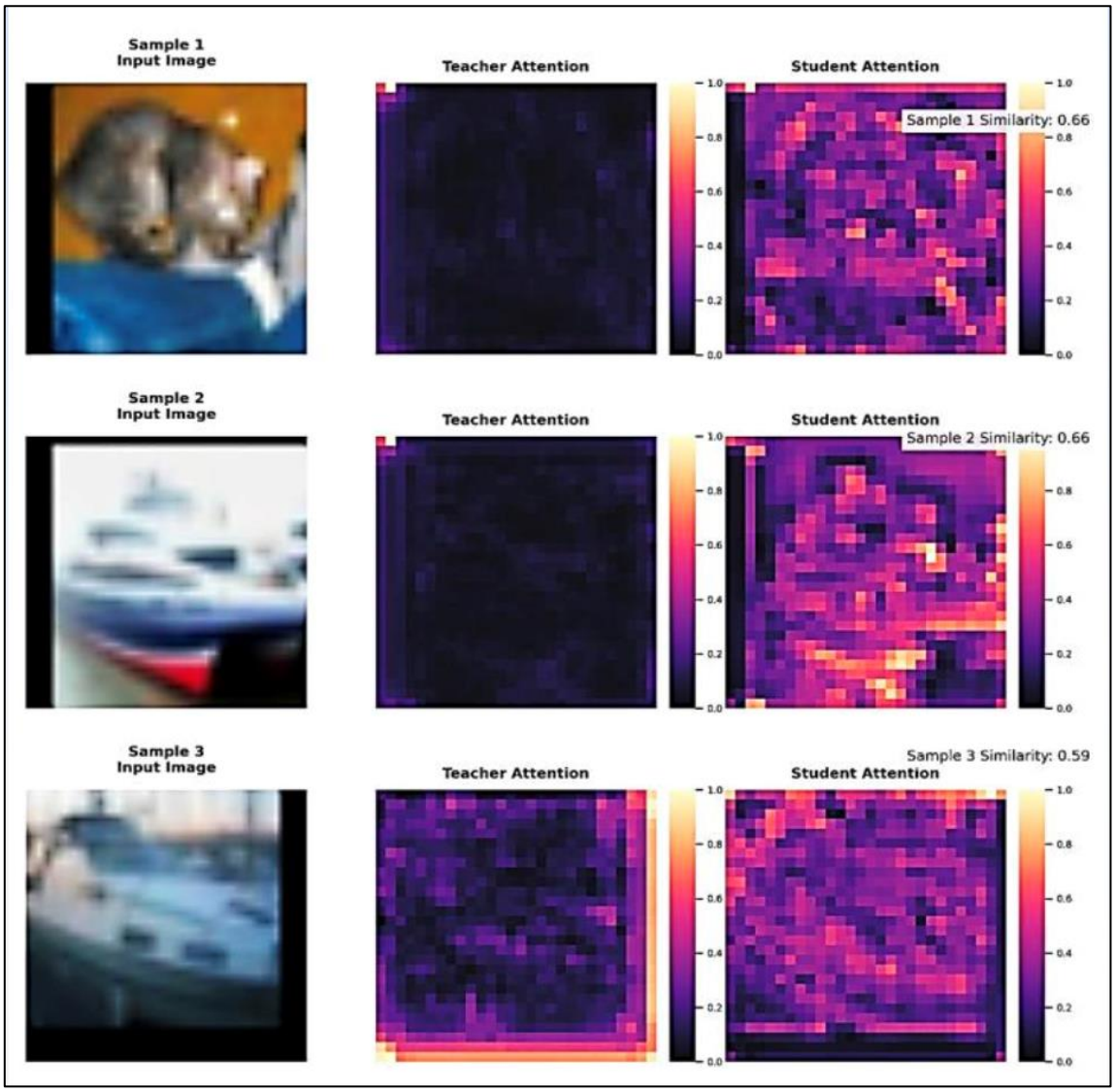


Figure 18: Attention map comparison for lower layer (layer 2) between Teacher and student

However, we noticed that the attention map of the student begins to align more closely with the attention map of the teacher, when the Attention Transfer is applied. As it can be seen in the figure 19, in later layers, the student zeroes in on the most discriminative regions like the ears on the animal with more consistency. Through these visual comparisons, it can be seen that the transfer of spatial attention cues between the teacher and the student helps the student a lot to imitate the critical focus areas of the teacher.

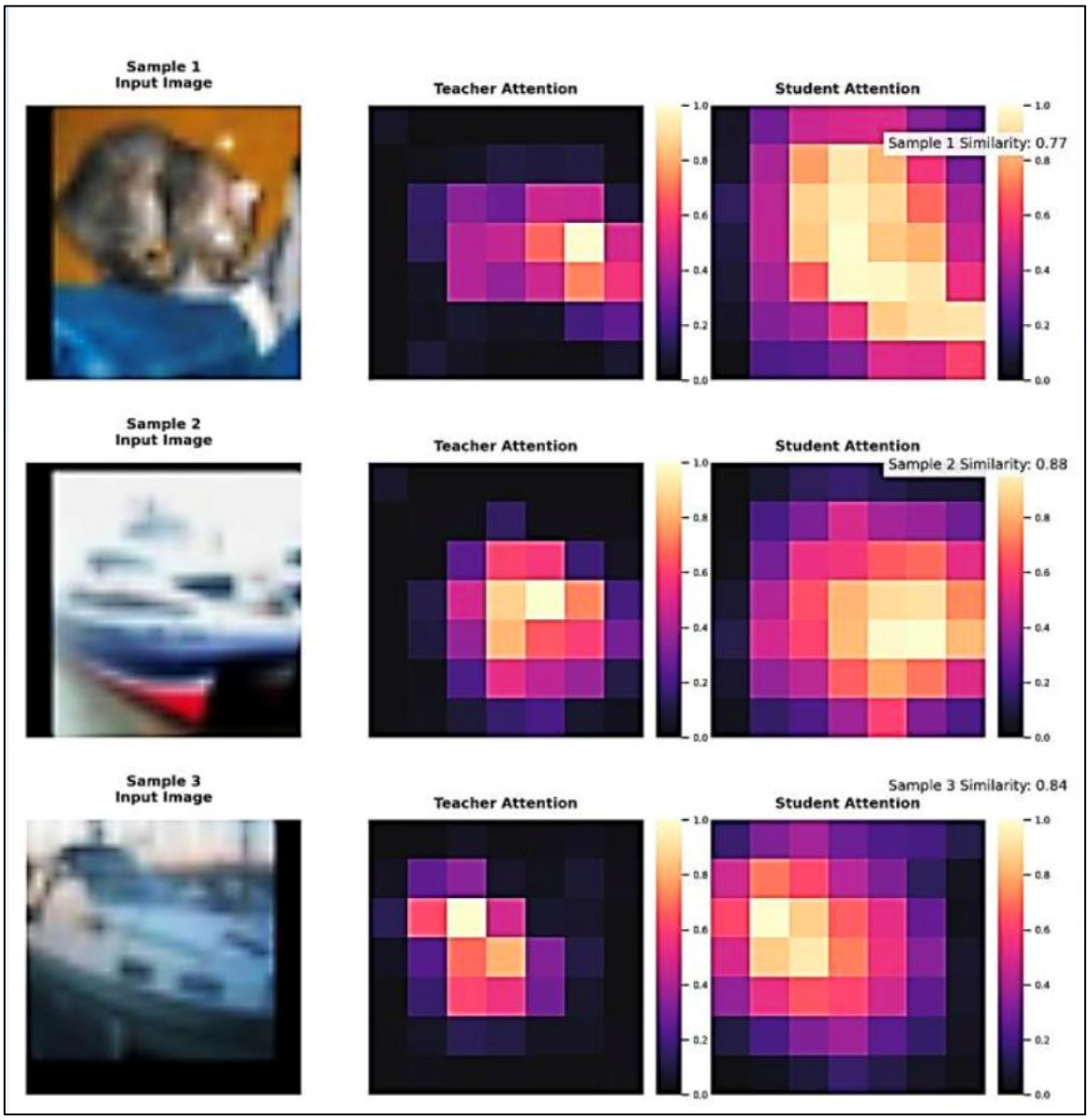


Figure 19: Attention map comparison for later layer (layer 4) between Teacher and student

LOSSES FOR MNIST, CIFAR-10, AND CELEBA

We have calculated various loss metrics across MNIST, CIFAR10, and CelebA to clearly understand how effectively our models are able to learn using Knowledge Distillation and Deep Mutual Learning. The loss metrics we used are -
i. KL Divergence (Lower KL Divergence means well matched probability distributions)
ii. Cross-Entropy (Lower Cross entropy means good performance)
iii. Cosine Similarity (Higher cosine similarity is better)

a. MNIST

i. KL Divergence

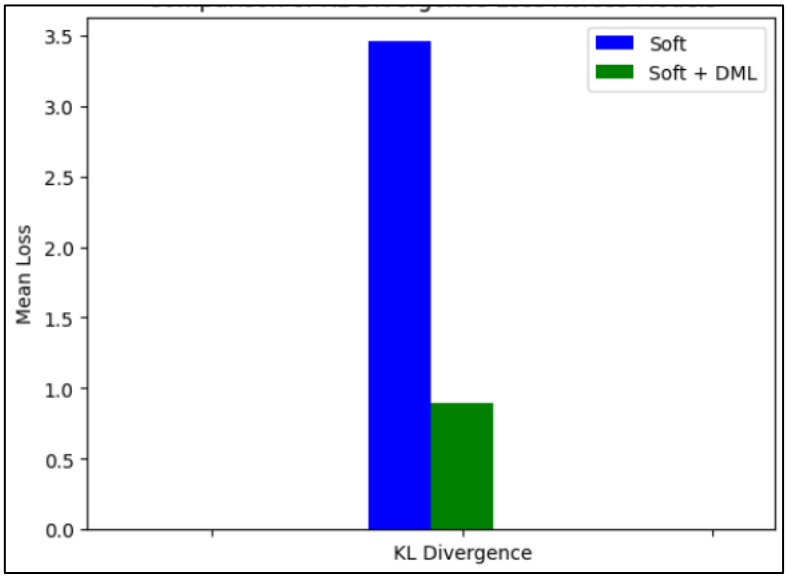


Figure 20: KL Divergence for Soft and soft + DML

The KL divergence drops very sharply for Soft Targets + DML. This is because the student is able to match the softened logits of the teacher very quickly. The simple digit patterns of the MNIST are the main reason for this alignment. The student can even surpass the accuracy of the teacher, once it is able to pick up the probability distributions of the teacher's class. This shows the synergy of collaborative learning on easier dataset.

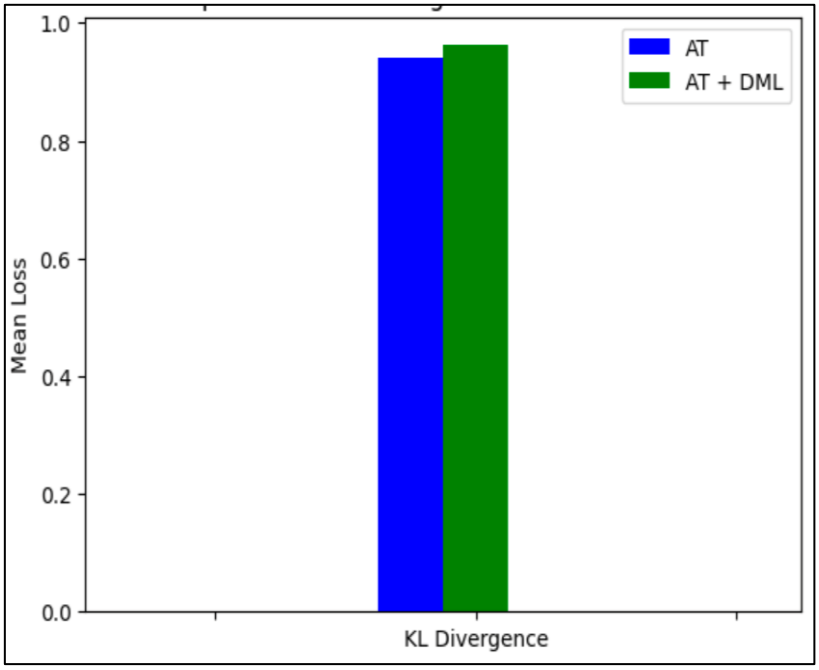


Figure 21: KL Divergence for AT and AT + DML

Here, in figure 21 it can be seen that, the KL divergence for Attention Transfer + DML is very slightly higher than that of Attention Transfer alone, demonstrating that mutual learning has maintained the alignment of the student towards the teacher's logits even after application of DML.

ii. Cross Entropy Loss:

Due to the simplicity of MNIST (as it does not demand intricate features), there is a minimal difference in CE loss for attention transfer. Whereas for soft target, we notice a drop in cross entropy loss which suggests that Mutual learning has pushed the student model closer to ground truth.

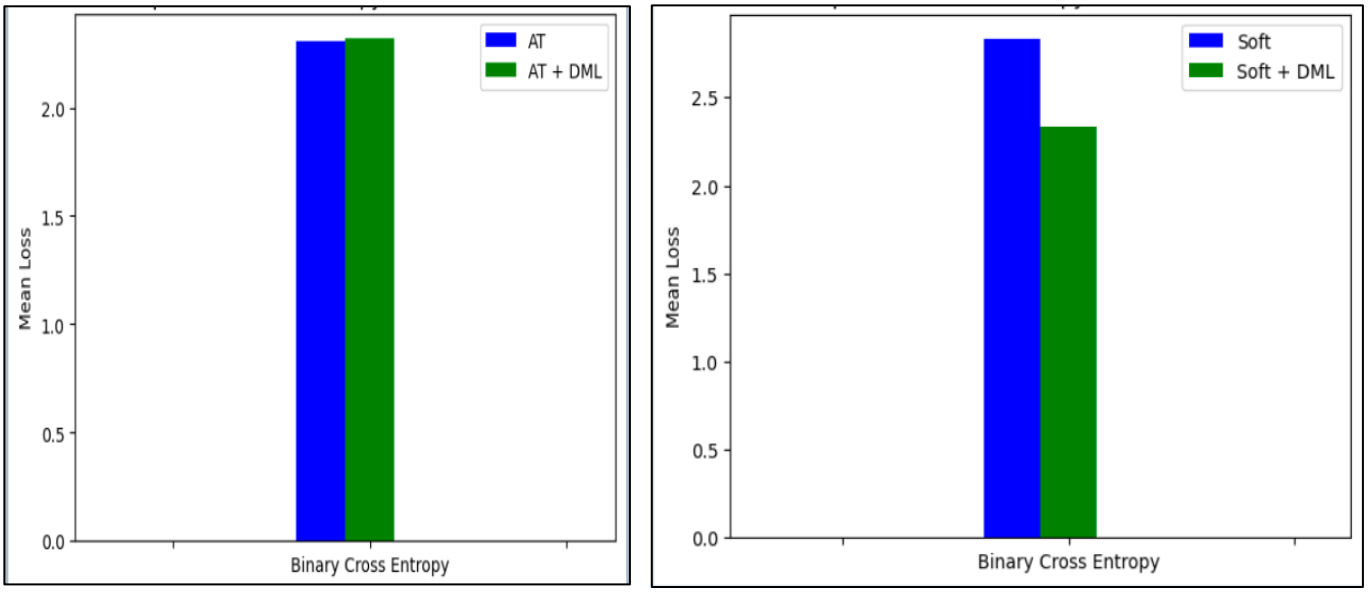


Figure 22: Cross Entropy Loss Comparison

iii. Cosine Similarity

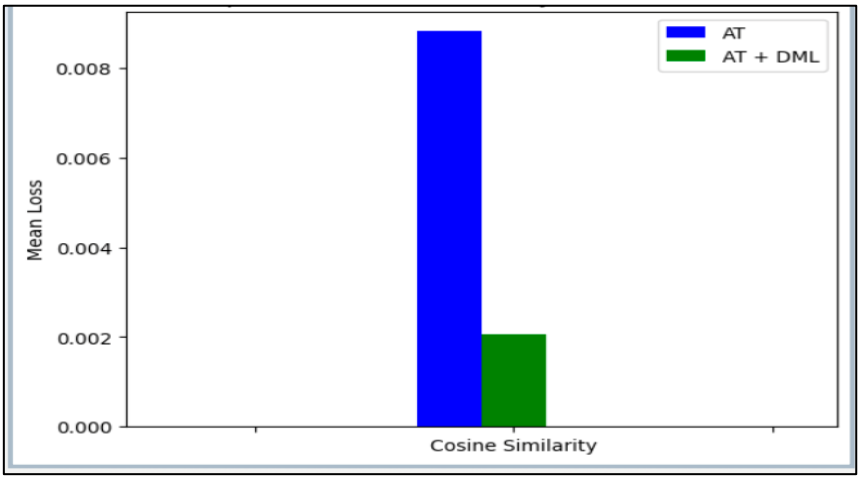


Figure 23: Cosine Similarity

In attention Student, we observe a drop of cosine similarity after mutual learning. This might be due to the fact that the feature Map representations learned from attention transfer is being traded off for boost in accuracy. After DML model focused on getting its predictions to match as close as possible to that of its peer and ground truth labels.

b. CIFAR-10

i. KL Divergence

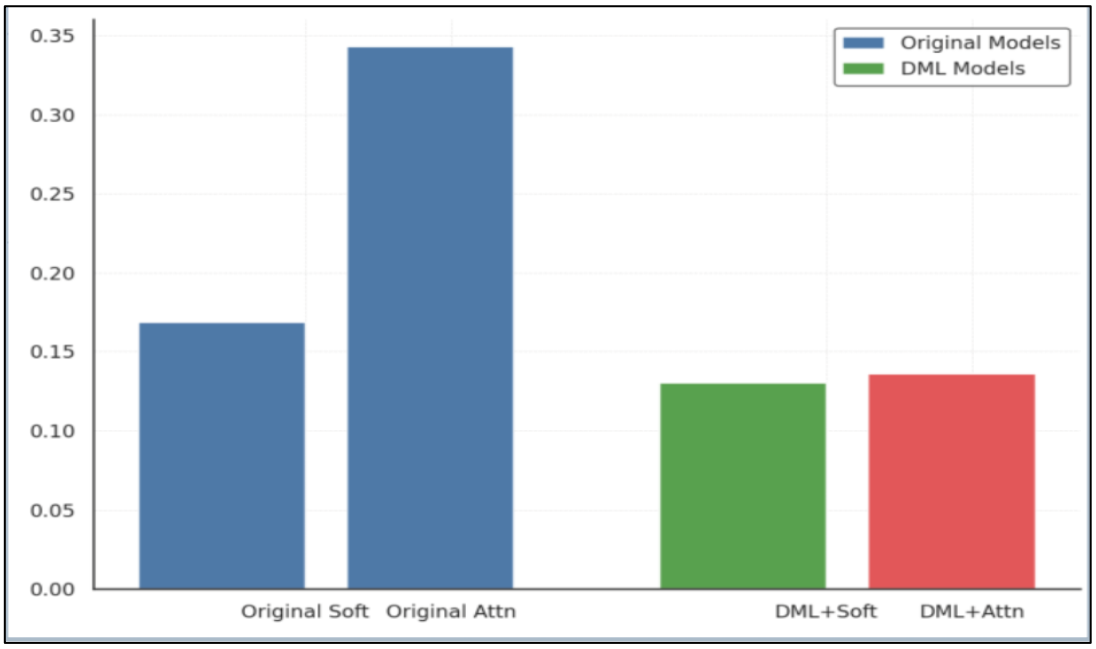


Figure 24: Kl divergence from teacher

In figure 24, the bar graph shows that Soft Target distillation gave slightly lower KL Divergence when compared to attention transfer. But it can be seen that after DML, The KL divergence of both student models, dropped quite a bit, with attention student experiencing the largest drop, indicating that it benefited quite a lot with DML.

ii. Cross Entropy Loss

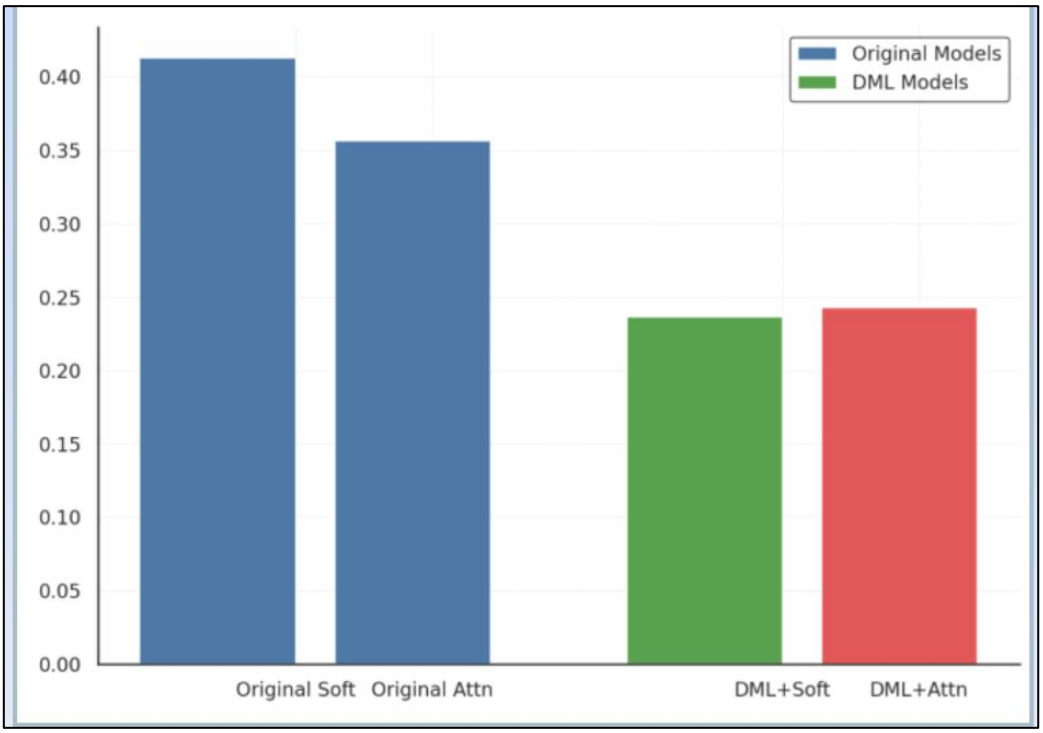


Figure 25: Cross Entropy Loss Comparison

It shows that in the original setups, for cross entropy loss, attention transfer student is better because it achieves a lower loss than soft target distillation. This suggests that in CIFAR-10, The attention student which learned teachers' representation is better in terms of predicting correct class labels. This bar graph also shows that DML improves performance of both distillation techniques.

iii. Similarity Heatmap

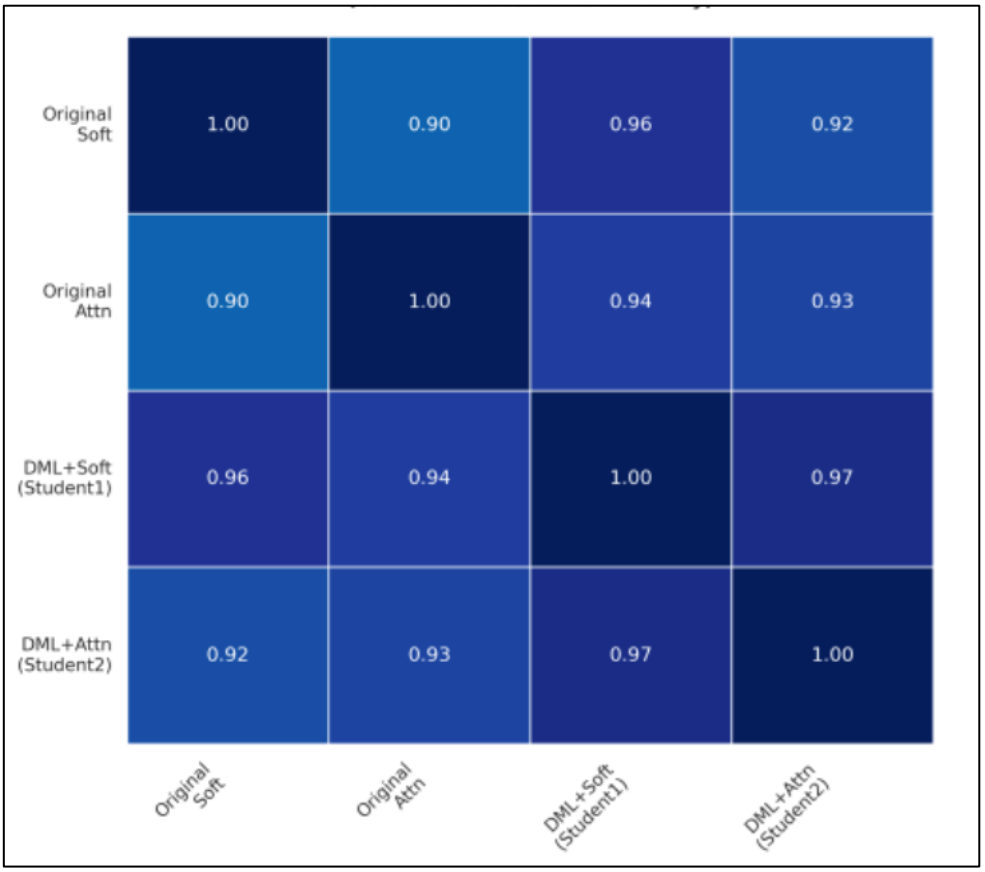


Figure 26: Model prediction similarity matrix

The above matrix highlights closer match between student and teacher predictions under DML, reflecting enhanced consistency in class outputs.

c. CelebA

1. Soft Targets + DML:

i. A very sharp drop in KL Divergence can be seen after DML. This indicates that DML student is able to mimic the output of the teacher closely when compared to original soft student.

ii. On the other hand, BCE Loss worsens slightly. This indicate that, the emphasis of the student on matching the logits of the teacher can sometimes lead to misalignment with the true label. Needs to be further investigated as to if the loss occurred due to multi label nature of dataset.

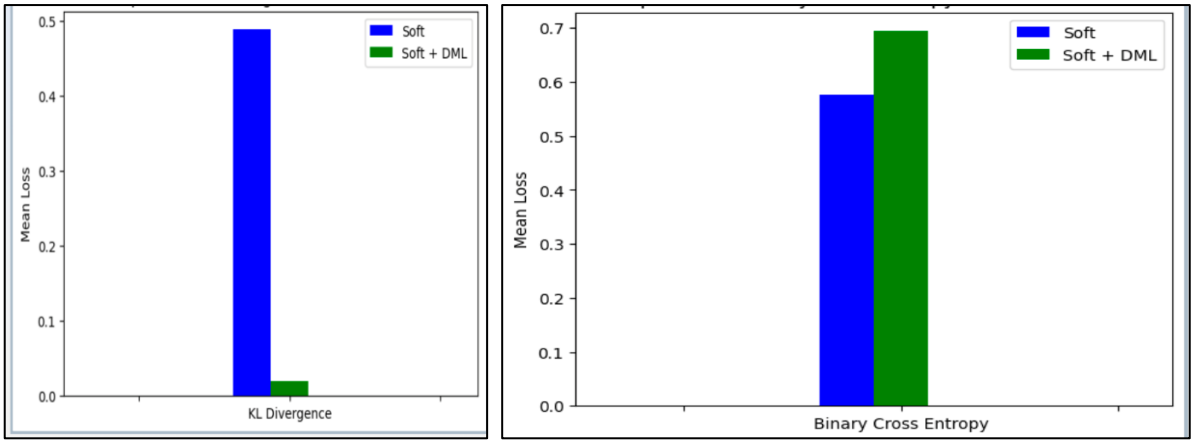


Figure 27: Comparison of KL divergence and binary cross entropy for Soft Target + DML

2. Attention Transfer + DML

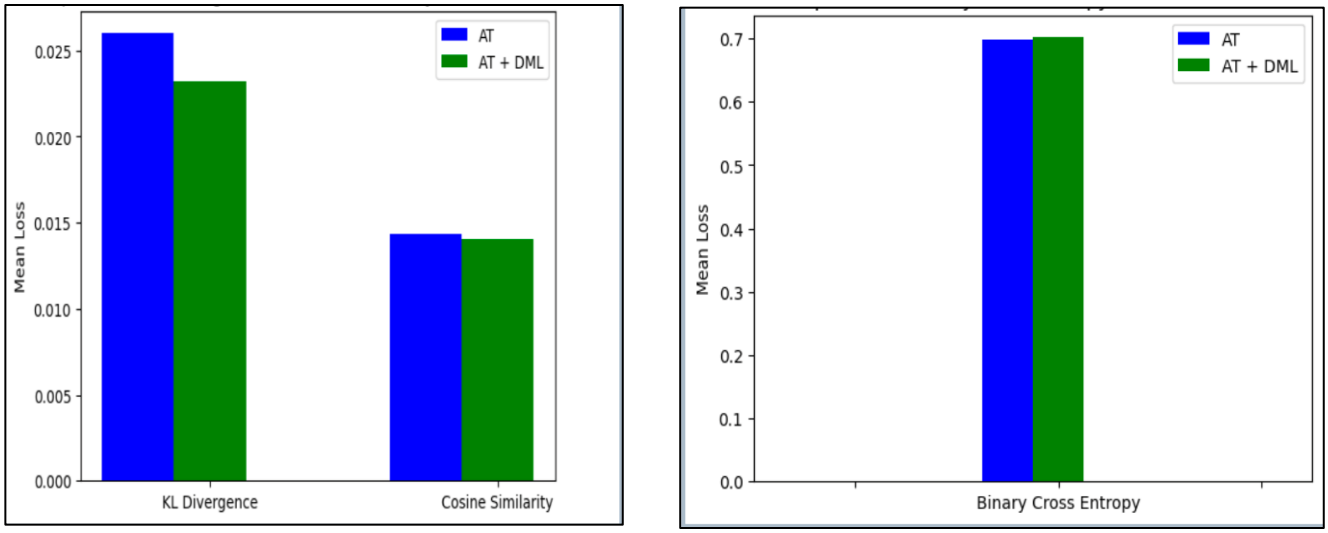


Figure 28: Comparison of KL divergence and binary cross entropy for Attention Transfer + DML

i. Here, there is a moderate reduction in KL divergence. This reflect that the approach is balanced because it still aligns the prediction of the teacher and student.

ii. BCE loss is stable. This indicates very strong facial attribute classification.

iii. Cosine Similarity is also stable which confirms that subtle attributes like eye shape, hair details are better captured because of feature alignment.

DEPLOYMENT

A Streamlit web application was developed to visualize and compare models trained using Knowledge Distillation (KD) techniques. The app is hosted at: https://teampurple.streamlit.app

The pipeline includes a ResNet-50 teacher model trained on CelebA and two student models, which then undergone DML:

- ResNet-18 trained with Soft Target
- ResNet-18 trained with Attention Transfer

Following teacher-student training, the students undergo Deep Mutual Learning (DML) for deployment efficiency.

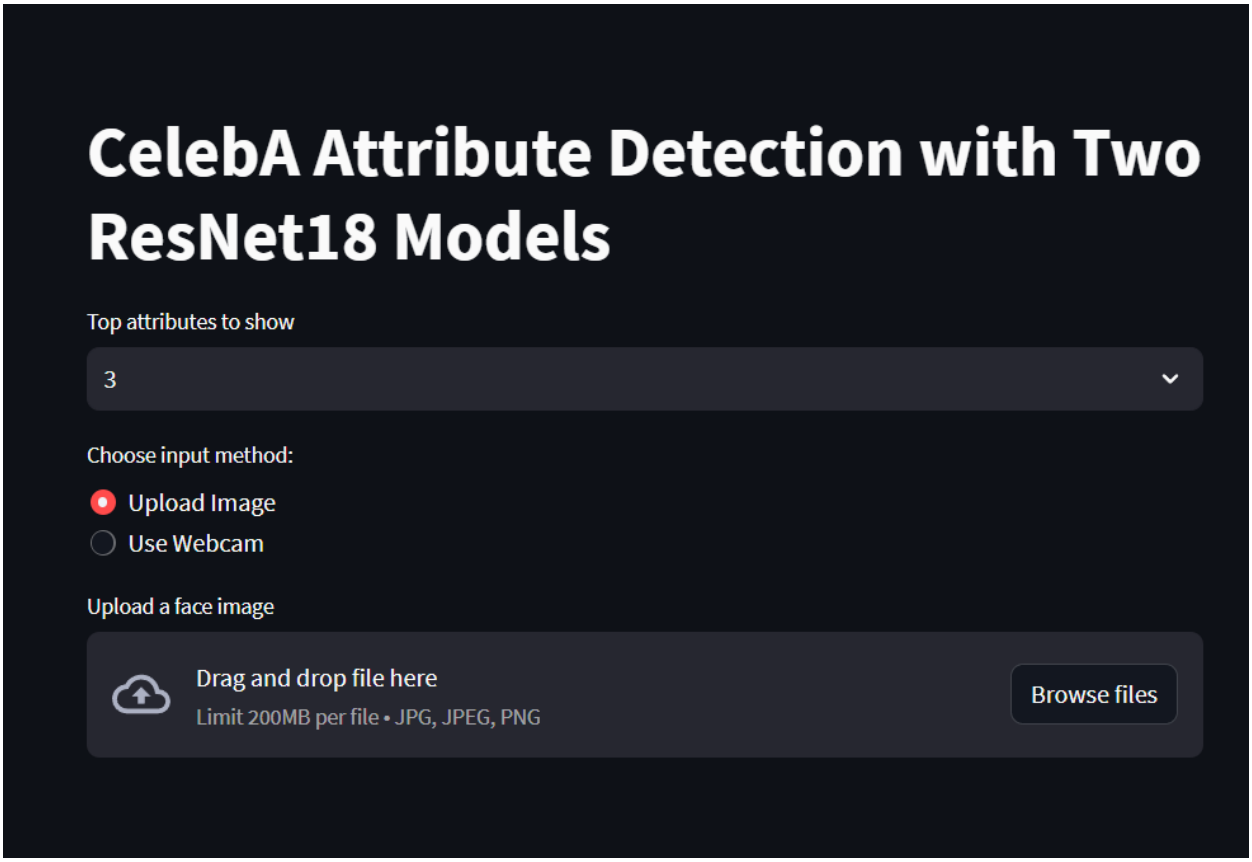


Figure 29: App Home Interface

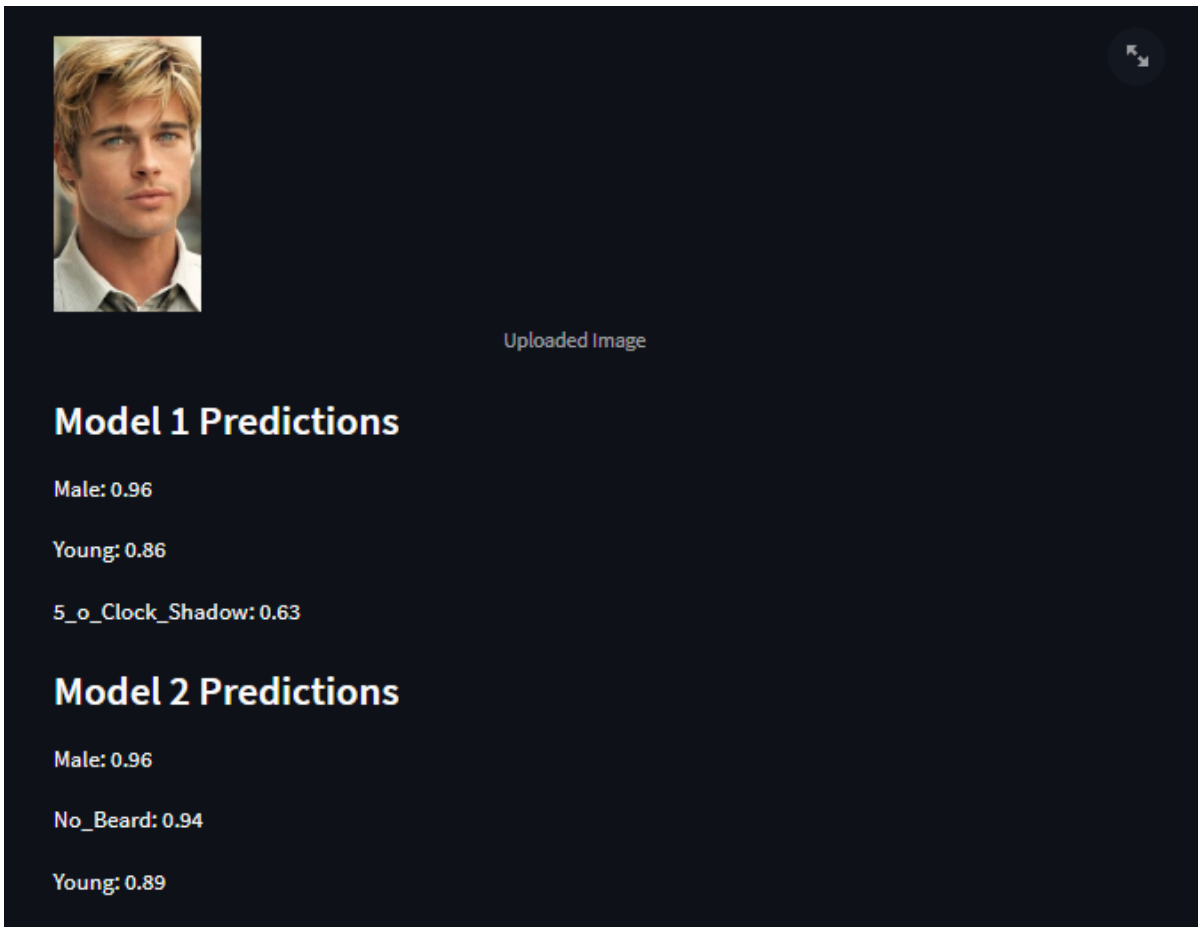


Figure 30: Student Predictions Comparison

CONCLUSION

Knowledge Distillation and Deep Mutual Learning when used together becomes a very powerful approach to design a system that has features of both high-capacity neural networks and more efficient and compact models. Through knowledge distillation, the student model gets a structured guidance from a teacher model, as a result of which it is able to easily learn complex representations. After this, when Deep mutual learning is applied, the multiple student models work together in cooperation and refine the predictions of each other, as a result of which more stronger and resilient feature learning occurs.

Through our experiments we found that when DML is applied on a more complex dataset like CelebA, then this results in enhanced generalisation in the student model. For a task like fine-grained facial attribute classification, this is very important, because here subtle details matter a lot. In case of simpler dataset like MNIST, substantial positive impact can be achieved even with small optimisations. In these scenarios, the student model is able to match or even surpass the accuracy of the teacher's model. Moreover, by combining KD and DML, we were able to compress more than half of the teacher model's size, without seeing significant loss in the accuracy.

Overall, we conclude that the combination of Knowledge Distillation and Deep Mutual Learning is effective on both simple and complex datasets. This shows that this combined method is effective and can be an efficient solution for real-world deep learning deployment.

REFERENCES

1) Hinton, Geoffrey E. et al. “Distilling the Knowledge in a Neural Network.” ArXiv abs/1503.02531 (2015): doi: 10.48550/arXiv.1503.02531
2) Y. Zhang, T. Xiang, T. M. Hospedales and H. Lu, "Deep Mutual Learning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4320-4328, doi: 10.1109/CVPR.2018.00454.
3) Zagoruyko, Sergey and Nikos Komodakis. “Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer.” ArXiv abs/1612.03928 (2016): n. pag.
4) Understanding BERT Variants: Part 2 - DistilBERT & TinyBERT using knowledge distillation https://medium.com/data-science-in-your-pocket/understanding-bert-variants-part-2-64ca0187373e
5) "Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer" https://github.com/szagoruyko/attention-transfer/tree/master?tab=readme-ov-file
6) Loss Functions in Deep Learning. https://www.geeksforgeeks.org/loss-functions-in-deep-learning/
7) CelebA - https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
8) CIFAR-10 - https://www.cs.toronto.edu/~kriz/cifar.html
9) MNIST - https://en.wikipedia.org/wiki/MNIST_database
10) Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation. https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f
11) Song, Huaxiang. (2023). A Consistent Mistake in Remote Sensing Images’ Classification Literature. Intelligent Automation & Soft Computing. 37. 1-18. 10.32604/iasc.2023.039315.