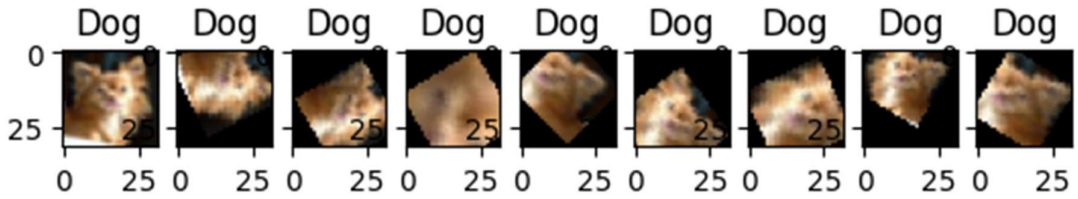
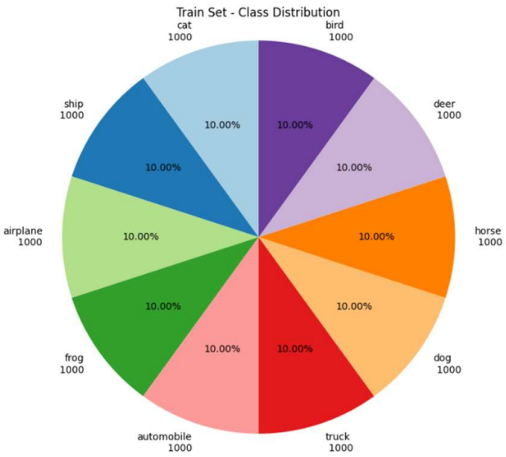
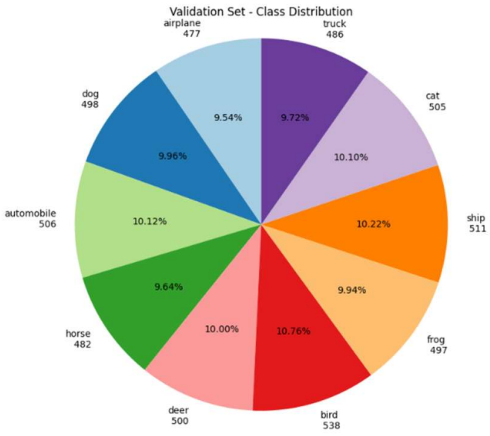
Part 1 – Exploration & insightful graphs

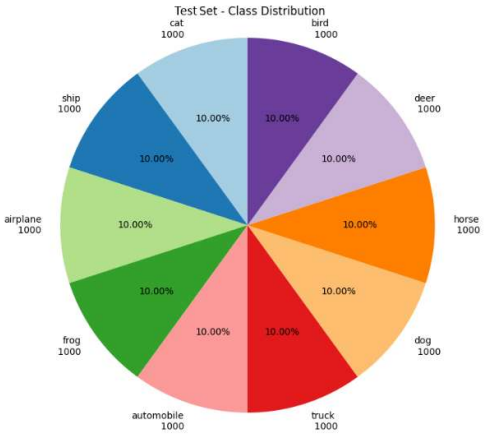
Initially, we wanted to examine the raw data and our augmentations. We used different methods to create different versions of the same image using crop, flip and affine.



Next, we wanted to examine the class distribution in a dataset. We chose this as it is a useful graph to understand if there is an imbalance in the data between classes.







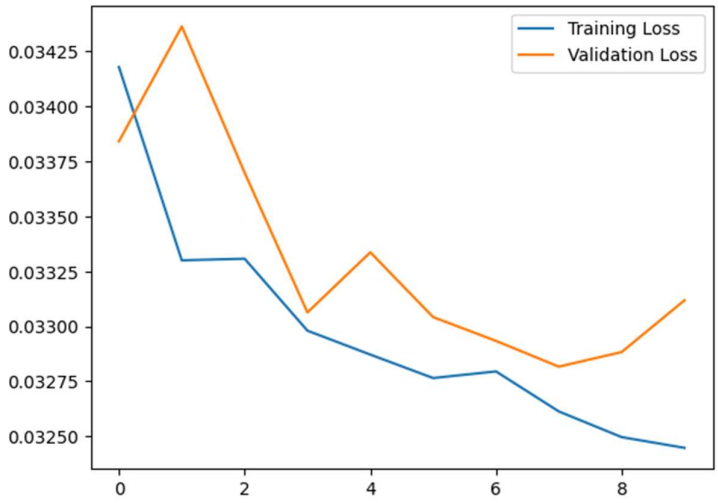
Since the validation set was created using a random sample from the train set it makes sense that the distribution is slightly imbalanced. On the other hand, the test set contains evenly distributed 10,000 samples as expected.

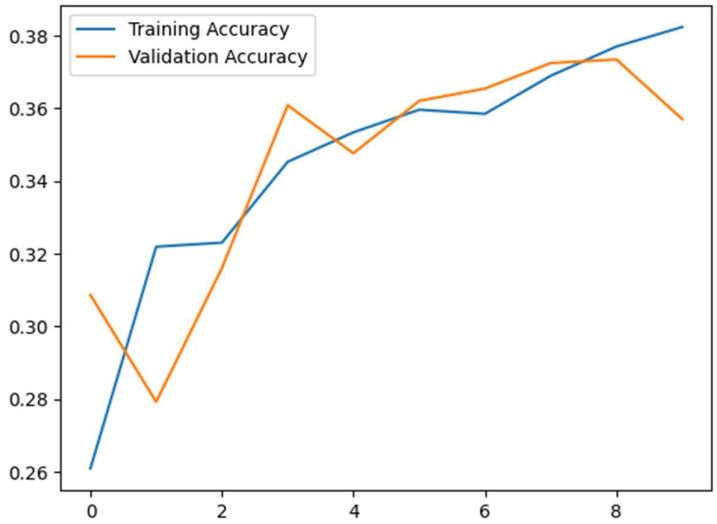
Part 2 – Training and Evaluation

1. VanillaMLP

We began by implementing a simple fully-connected neural network – Vanilla MLP – consisting only of linear layers and ReLU activations. We used the CIFAR10 dataset to train the model using various combinations of hyperparameters.

First experiment
Input size: 3*32*32 (will remain the same in all experiments)
Learning rate: 0.001
Batch size: 64
Hidden size: 512
# of hidden layers: 3
Epochs: 10





The training loss and the validation loss consistently decreased up to epoch #7 where the validation loss increased again, could be as a consequence of

overfitting. The training accuracy increased to ~0.38 However, the validation accuracy dropped to ~0.35 on the last 3 epochs.
More experiments are needed to further examine this theory and determine whether it was a result of an overfit or maybe the model had a minor setback that could have been optimized further. We concluded the model can use more epochs to further examine the process and maybe reach a plateau.

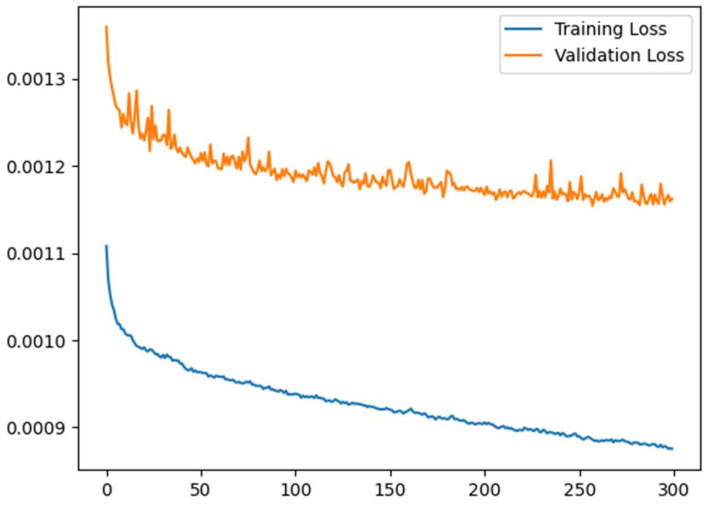
2. *Imporved VanillaMLP*

To further improve performance, we implemented a deeper and regularized version of the model with the goal of enhancing generalization and reducing overfitting. We added the following:

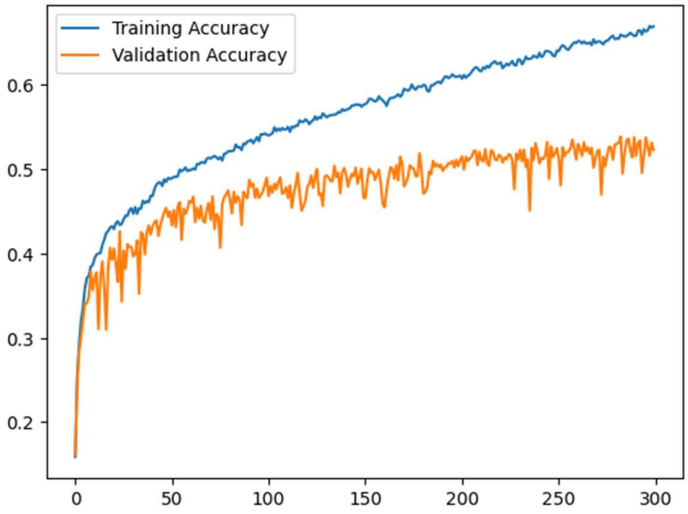
- Deeper Archtecture: Increased the number of hidden layers to 16, allowing the model to learm more complex representations.
- Batch normalization: applied after each linear transformation to stabilize and accelerate training.
- Dropout (rate=0.2): added after each activation layer to prevent overfitting by randomly deactivating neurons during training.
- Activation Function: Used ReLU for non-linearity after each hidden layer.
- Softmax Output: Applied at the end to produce probability distribution over the classes.
- Optimizer: Adam
- Loss Fun: CrossEntropyLoss

Best network
Input size = 3 * 32 * 32
Hidden size = 512
Num classes = 10
Num hidden layers = 16
Num epochs = 300
Dropout p=0.2
Transformation p=1
Learning rate = 0.001
Batch size = 2056

These hyperparameters were the result of different experiments. We "played" with the number of hidden layers, epochs batch size and hidden layers size. We even increased the dropout p. eventually we concluded the model worked best with a large batch size and a decreased hidden layer size. We chose to leave the learning rate the same.



Looking at the first graph, we can see that both the training loss and validation loss steadily decrease during the training process.
The training loss shows a smooth and consistent decline across 300 epochs. The validation loss decreases initially and then plateaus, with some small fluctuations, but maintains a stable downward trend overall. This suggests that the model is learning effectively and is not suffering from severe overfitting.
The training accuracy increases consistently throughout the training and reaches above 65%.



The validation accuracy improves significantly in the early epochs and stabilizes around 55% towards the end of training. The gap between training and validation accuracy indicates some level of overfitting, but it's relatively moderate.

After completing the training process, we evaluated the final model on the test set. The results were:
Test Loss: 1.9433,
Test Accuracy: 51.63%

These results show that the model was able to generalize accurately, slightly more than half of the time. But the accuracy is still relatively low for a classification task like CIFAR-10. We assume that the model has learned useful patterns in the training data, but not enough to achieve high generalization performance.

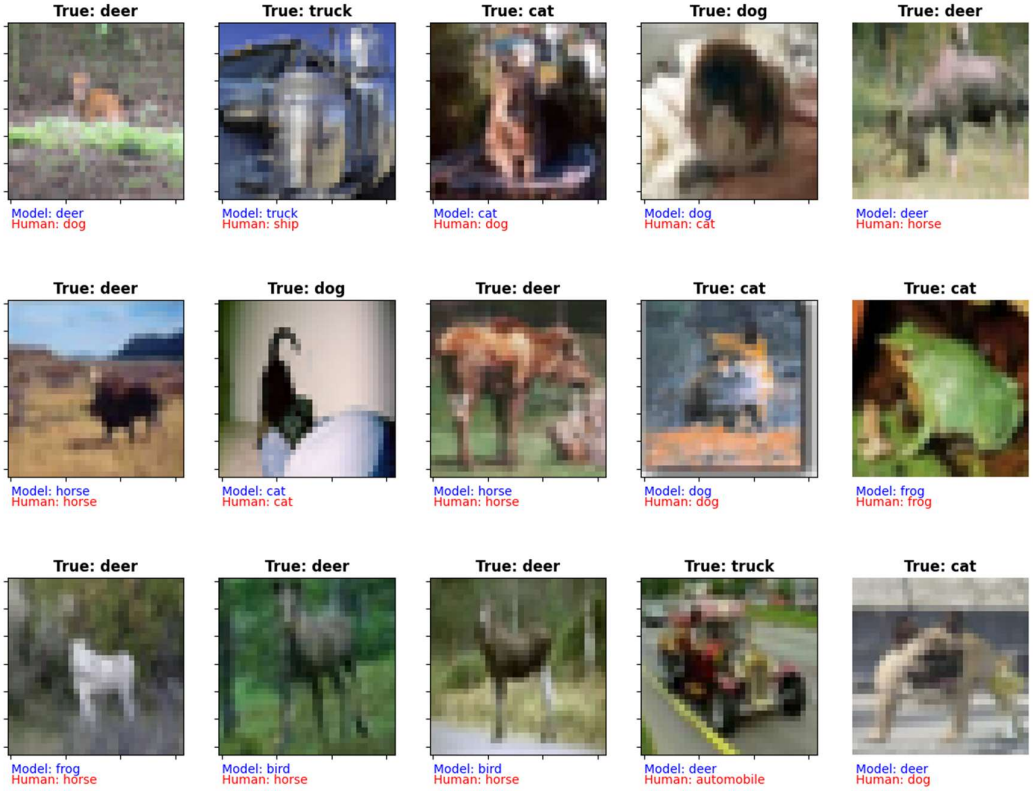
Part 3 – CIFAR10-human

In this section, we compare the model's predictions with human-derived labels from the CIFAR-10H dataset. This helps assess how closely the model's confidence aligns with human uncertainty.

Q: Which group did the model agree with more: the humans or the true labels? Why do you think that is the case?

A: Interestingly, the model agreed with both the true labels and the human labels in 22 out of the 79 mismatches.

Example images tagged by the true label, the model and the human label.



Q: Are there any interesting insights you can extract from the graph?

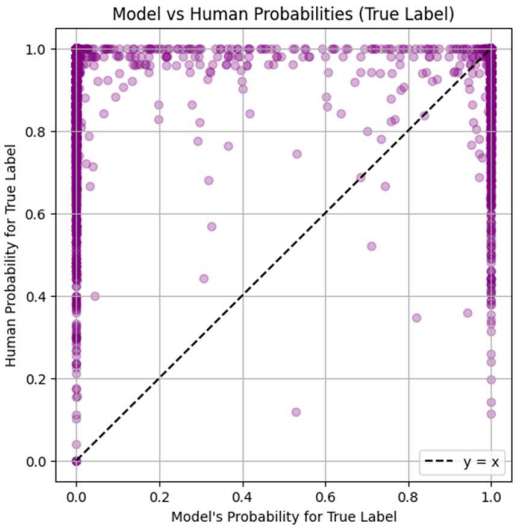
A: Examining the example images, we can observe various patterns of agreement between the three labels. In the first image on the left, the model correctly identified the animal as a deer, while the human misclassified it as a dog. Although this mistake is understandable, the model performed more accurately, possibly due to differences in neural network processing compared to the human brain.

In other cases, we see discrepancies across all three labels. For instance, in the last image, neither the human nor the model correctly identified the cat—though each made a different error. While we can intuitively grasp why humans might mislabel certain images, deciphering the model's decision-making process is far more complex. Despite experimenting with various hyperparameter configurations, fully

understanding the reasoning behind its predictions remains elusive and may even be impossible.

Interestingly, the model mirrors human errors in several instances. Among the 79 mismatches observed, the model made the same mistakes as humans in 22 cases, while correctly identifying 22 images that the human misclassified. These patterns suggest intriguing parallels between human perception and machine learning accuracy.

To evaluate how closely the model's predictions align with human judgment, we plotted a scatter plot comparing the model's probability for the true label (x-axis) against the average human probability for the true label (y-axis).



This scatter plot reveals intriguing patterns in the confidence levels of both the model and human labelers. Many data points cluster around the corners (0,1) and (1,1), indicating that when humans are highly confident in the correct label, the model either strongly agrees (probability near 1) or strongly disagrees (probability near 0).
The presence of vertical bands at $x \approx 0$ and $x \approx 1$ suggests that the model frequently makes high-confidence predictions, either being entirely correct or entirely incorrect. While this shows some alignment between human and model decisions, significant discrepancies remain—likely in ambiguous or more challenging cases.

Ultimately, this graph illustrates that the model and human perceive patterns differently. While humans rely on intuition and experience, the model follows mathematical representations that can sometimes lead to unexpected errors. These differences highlight the need for further analysis to understand where and why the model deviates from human judgment.

Part 4 – Bonus

Q: Is there a correlation? In other words, does the model have difficulties with images for which human subjects had high reaction times (took time to label)?

A: Correlation between Model Probabilities and Reaction Time: We calculated a Pearson correlation coefficient of **0.014** with a p-value of **0.165** between the model's probabilities for the true label and the human reaction times. This **low correlation coefficient and the high p-value (above 0.05) suggest that there is no strong statistically significant linear relationship between the model's confidence in its correct predictions and the time it took humans to label the same images** [implied from our conversation history regarding the interpretation of correlation metrics]. In other words, the model wasn't necessarily less sure about images that humans took longer to label, or vice versa.

Q: Are the images with high reaction times also images that the human subjects mislabeled? Did the model label them the same as the humans, or did it predict the correct label regardless?

A:
Human Accuracy and High Reaction Time Images: The average human accuracy we found was **0.992**, which is very high. However, we identified **79 images that were mislabelled by humans**, and a significant portion of these (**74 images**) had **high reaction times**. This finding **indicates a connection between longer human deliberation times and a higher likelihood of making labelling errors** [implied from our conversation history analysing this specific finding]. It appears that when humans struggled more with an image (took longer to respond), they were also more prone to misclassifying it. Part 3 of the assignment, as mentioned in ["Assignment 1.pdf": 2], focused on the CIFAR10-Human dataset where these human labelling experiments were conducted, allowing for this type of analysis.

Model Performance on High Reaction Time Images: The agreement rate of our model with the human labels on high reaction time images was **0.392**, and the model's accuracy on these same images was **0.393**. These low rates suggest that **our neural network model also struggled with the images that humans found difficult (those with high reaction times)**.
The fact that both humans and the model performed poorly on these images indicates that these images might inherently be more challenging to classify. However, the slight difference between the agreement rate with human labels (0.392) and the model's accuracy (0.393) implies that **the model did not necessarily make the same mistakes as the humans** on these difficult images. While both struggled, some of their incorrect predictions have differed. Our best-performing model from Part was used for this comparison.