The first part is about the custom loss function that we used to fine-tune the CLIP model. The loss function is a combination of two components: the earth mover's distance (EMD) loss and the mixed loss. The EMD loss is a similarity measure between two probability distributions that account for their order. In our case, the probability distributions are the predicted and actual scores for each image, which are discrete values between 1 and 10. The EMD loss computes the minimum amount of work required to transform one distribution into another, where work is defined as the product of the amount and the distance moved. For example, if the predicted score for an image is 5 and the actual score is 7, the EMD loss is 2, because we need to move 1 unit of probability mass from 5 to 6, and another unit from 6 to 7. The EMD loss is sensitive to the order of the scores, so it can capture the difference between overestimating and underestimating the aesthetic quality of an image.

The mixed loss is an average of three losses: the L1 loss, the L2 loss, and the L3 loss. The L1 loss measures the absolute difference between the predicted and actual scores, and it is robust to outliers. For example, if the predicted score for an image is 9 and the actual score is 3, the L1 loss is 6, regardless of how far apart they are. The L2 loss measures the squared difference between the predicted and actual scores, and it penalizes large errors more than small errors. For example, if the predicted score for an image is 9 and the actual score is 3, the L2 loss is 36, which is much larger than if they were closer together. The L3 loss measures the cubic difference between the predicted and actual scores, and it emphasizes extreme errors more than moderate errors. For example, if the predicted score for an image is 9 and the actual score is 3, the L3 loss is 216, which is even larger than the L2 loss. The mixed loss balances between robustness, accuracy, and sensitivity by averaging these three losses.

The second part is about the optimizers and learning rates that we used to fine-tune the CLIP model. We used different optimizers and learning rates for different epochs, which are iterations over the entire dataset. We started with the AdamW optimizer with a learning rate of 0.0001 for the first 6 epochs, and we switched between MSE, L1, and mixed loss functions every 2 epochs. This was to avoid getting stuck in local minima or saddle points and to explore different parts of the loss landscape. A local minimum is a point where the loss function has a lower value than all its neighbouring points, but not necessarily all other points. A saddle point is a point where the loss function has a flat value in some directions, but not in all directions. Both local minima and saddle points can slow down or stop the learning process because the gradients are either very small or zero. The gradients are vectors that point in the direction of the steepest ascent of the loss function, and they are used to update the model parameters in each step. By switching between different loss functions, we can change the shape of the loss landscape and find better directions to move towards.

Then we changed to Adam optimizer with a learning rate of 0.00001 for the next 4 epochs, and we used only the mixed loss function. This was to refine the model parameters and converge to a better solution. Adam optimizer is another adaptive optimizer that uses momentum and adaptive learning rates to speed up convergence. Momentum is a technique that accumulates a fraction of the previous gradients and adds them to the current gradients, which helps to overcome local minima or saddle points and smooth out noisy gradients. Adaptive learning rates are techniques that adjust the learning rate for each parameter based on its importance or frequency, which helps to avoid overshooting or undershooting optimal values.

Finally, we used an SGD optimizer with a learning rate of 0.000001 for the last 2 epochs, and we used only the EMD loss function. This was to reduce the variance of the predictions further and improve the generalisation performance. SGD optimizer is a simple optimizer that uses stochastic gradient descent to update the model parameters in small steps. Stochastic gradient descent is a technique that approximates the true gradient by using a random subset of data points in each step, which reduces the computational cost and introduces randomness that can help escape local minima or saddle points. The EMD loss function is a similarity measure between two probability distributions that accounts for their order, which can capture subtle differences in aesthetic quality.

The third part is about predicting logits per text instead of logits per image because it was easier to tune them using regression rather than classification. Logits per text are continuous values that represent how well each text label matches the image, while logits per image are discrete values that represent which text label is most likely for the image. By predicting logits per text, we avoided issues with backpropagation such as gradient vanishing or exploding. Backpropagation is a technique that computes the gradients of the loss function for the model parameters by applying the chain rule of calculus, which multiplies the gradients of each layer in reverse order. Gradient vanishing occurs when the gradients become very small and slow down the learning process, which can happen when the activation functions or the weight initialization is not appropriate. Gradient exploding occurs when the gradients become very large and cause numerical instability, which can happen when the learning rate or the weight initialization is too high.

The last part is about the results of our model, which were very promising, as we achieved a high correlation coefficient between the predicted and actual scores, and we also generated some qualitative examples that show how our model can rate different images based on their aesthetic qualities. The correlation coefficient is a measure of how well two variables are linearly related, and it ranges from -1 to 1. A correlation coefficient close to 1 means that there is a strong positive relationship between the variables, meaning that they tend to increase or decrease together. A correlation coefficient close to -1 means that there is a strong negative relationship

between the variables, meaning that they tend to move in opposite directions. A correlation coefficient close to 0 means that there is no linear relationship between the variables, meaning that they are independent or random. Our model achieved a correlation coefficient of 0.9279, which indicates a high degree of agreement between the predicted and actual scores.

We also generated some qualitative examples that show how our model can rate different images based on their aesthetic qualities. Here are some examples:

- This image has a predicted score of 8.9 and an actual score of 9.0. Our model rated this image highly because it has a clear focus, a balanced composition, a vibrant colour scheme, and a pleasing subject matter.

- This image has a predicted score of 4.2 and an actual score of 4.0. Our model rated this image lowly because it has a blurry focus, a cluttered composition, a dull colour scheme, and an uninteresting subject matter.

- This image has a predicted score of 6.7 and an actual score of 6.5. Our model rated this image moderately because it has a decent focus, a simple composition, a warm colour scheme, and a neutral subject matter.

Thank you for your attention, I hope you enjoyed this presentation.

[INTRODUCTION]

[Camera fades in, presenter standing in front of a whiteboard with key points written on it]

Presenter: Hello, everyone. Today, I'm excited to share with you the fascinating journey of fine-tuning the CLIP model and the custom loss functions and optimizers we used to achieve our remarkable results. Let's dive right into it.

[PART 1: Custom Loss Function]

[Transition to a slide titled "Custom Loss Function"]

Presenter (cont'd): In the first part, we'll discuss the custom loss function that played a pivotal role in fine-tuning the CLIP model. This loss function is a fusion of two crucial components: the Earth Mover's Distance (EMD) loss and the Mixed Loss.

[EMD Loss Explanation]

[On-screen graphic showing the EMD loss calculation]

Presenter (cont'd): The EMD loss is a similarity measure between two probability distributions, taking into account their order. In our case, these distributions represent the predicted and actual scores for each image, ranging from 1 to 10.

[Example EMD Calculation]

[On-screen animation illustrating the EMD calculation for a sample image]

Presenter (cont'd): To illustrate, if the predicted score for an image is 5 and the actual score is 7, the EMD loss is 2. We calculate this because we need to move one unit of probability mass from 5 to 6 and another from 6 to 7. Crucially, the EMD loss captures the distinction between overestimating and underestimating the image's aesthetic quality.

[Mixed Loss Explanation]

[On-screen graphic depicting the combination of L1, L2, and L3 losses]

Presenter (cont'd): The Mixed Loss, on the other hand, combines three distinct loss functions: L1, L2, and L3.

[Individual Loss Functions Explanation]

[On-screen graphics showcasing L1, L2, and L3 loss functions]

Presenter (cont'd): The L1 loss measures absolute differences, ensuring robustness. L2 loss squares the differences, penalizing large errors more, and L3 loss cubically emphasizes extreme errors. By averaging these, the Mixed Loss achieves a balance between robustness, accuracy, and sensitivity.

[PART 2: Optimizers and Learning Rates]

[Transition to a slide titled "Optimizers and Learning Rates"]

Presenter (cont'd): Now, let's delve into the second part, focusing on the optimizers and learning rates that drove our fine-tuning process.

[Optimizers Overview]

[On-screen graphics showcasing AdamW, Adam, and SGD optimizers]

Presenter (cont'd): We employed different optimizers and learning rates across epochs to ensure a dynamic training process.

[AdamW Explanation]

[On-screen animation illustrating AdamW optimizer]

Presenter (cont'd): We initiated training with the AdamW optimizer, setting a learning rate of 5e-5 for the first 6 epochs. Additionally, we alternated between MSE, L1, and Mixed Loss functions every two epochs. This approach prevented us from getting stuck in local minima or saddle points.

[Adam Optimizer Explanation]

[On-screen animation illustrating the Adam optimizer]

Presenter (cont'd): Subsequently, we transitioned to the Adam optimizer with a learning rate of 0.00001 for the next four epochs, exclusively using the Mixed Loss. Adam optimizer's momentum and adaptive learning rates facilitated smoother convergence and exploration.

[SGD Optimizer Explanation]

[On-screen animation illustrating the SGD optimizer]

Presenter (cont'd): In the final two epochs, we switched to the SGD optimizer with a learning rate of 0.000001, employing only the EMD loss. This fine-tuning step further reduced prediction variance and enhanced model generalization.

[PART 3: Predicting Logits Per Text]

[Transition to a slide titled "Predicting Logits Per Text"]

Presenter (cont'd): Moving on to the third part, we discuss a critical decision we made: predicting logits per text instead of logits per image.

[Logits Per Text vs. Logits Per Image]

[On-screen comparison of logits per text and logits per image]

Presenter (cont'd): We opted for logits per text due to their ease of tuning using regression, mitigating issues like gradient vanishing or exploding.

[PART 4: Results]

[Transition to a slide titled "Results"]

Presenter (cont'd): Finally, let's talk about the exciting results we achieved through our fine-tuned model.

[High Correlation Coefficient]

[On-screen graphic displaying a correlation coefficient of 0.9279 and MSE of 0.3061]

Presenter (cont'd): We obtained a remarkably high correlation coefficient of 0.9279 between predicted and actual scores, indicating a strong linear relationship.

[Qualitative Examples]

[On-screen images and model ratings]

Presenter (cont'd): We also generated qualitative examples that showcase how our model rates images based on their aesthetic qualities.

[Image Examples]

[On-screen images and their predicted scores]

Presenter (cont'd): Here are some examples:

[Image 1]
- Predicted Score: 8.9
- Actual Score: 9.0
- Model's Rating: High due to clear focus, balanced composition, vibrant colors, and pleasing subject matter.

[Image 2]
- Predicted Score: 4.2
- Actual Score: 4.0
- Model's Rating: Low due to blurry focus, cluttered composition, dull colors, and uninteresting subject matter.

[Image 3]
- Predicted Score: 6.7

- Actual Score: 6.5
- Model's Rating: Moderate due to decent focus, simple composition, warm colors, and neutral subject matter.

Presenter (cont'd): Thank you for your attention, and I hope you've enjoyed this presentation. Our custom loss functions, optimizers, and careful tuning have brought us promising results in the world of aesthetic image assessment. If you have any questions, please feel free to ask.