Overfitting issues in this project

* Overfitting occurs when the model performs well on the training data but poorly on testing data.
* In the provided code, we monitor the training and testing loss and accuracy over epochs.
* If the training loss continues to decrease while the testing loss increases or remains stagnant, it indicates overfitting.
* Similarly, if the training accuracy keeps improving while the testing accuracy plateaus or decreases, it suggests overfitting.
* In our model, training loss started from 1.012 and kept decreasing till 0.370 after 10 epochs. Similarly, accuracy increases from 68.8338 to 83.172 after 10 epochs.
* But, for testing data, loss is increasing and accuracy is decreasing which shows that we are facing overfitting issues in this project.

Handling Overfitting:

* Dropout layers are used to address overfitting by introducing randomness during training, preventing the network from relying too much on specific features or neurons.
* Increasing the dropout rate (e.g., from 0.2 to 0.3) after the convolutional layers can further regularize the network, potentially reducing overfitting.
* Experimentation with different dropout rates and other regularization techniques (e.g., L2 regularization) can help mitigate overfitting.
* Yes, overfitting will reduce if we use another dropout layer (with rate 0.3) as can be seen in the difference in loss and accuracy.

Share Structure Property and Invariance Property:

* Share structure property refers to the idea that certain features learned by the model can be shared across different parts of the input space. In CNNs, weight sharing across convolutional layers enables the model to learn spatial hierarchies of features, capturing patterns and structures that are invariant to translation.
* Invariance property refers to the network's ability to recognize patterns regardless of their location in the input. This is achieved through convolutional and pooling layers, which extract features and down-sample spatial dimensions, making the model robust to translations, rotations, and other transformations in the input data.
* In the project, the CNN architecture leverages share structure property through weight sharing in convolutional layers and exploits invariance property through max pooling layers, enabling the model to learn hierarchical representations of sign language letters invariant to variations in hand gestures' positions and orientations.