1. Model architecture, optimization method andparameters

1.1 Modelarchitecture

(1) **Convolutionlayer1**	64 channels, k = 4, s = 1, P = 2*(with batchnormalization)*
(2) Convolutionlayer2	64 channels, k = 4, s = 1, P =2
(3) MaxPooling	s = 2, k =2
(4) Dropout	r =0 . 1
(5) **Convolutionlayer3**	64 channels, k = 4, s = 1, P = 2*(with batchnormalization)*
(6) Convolutionlayer4	64 channels, k = 4, s = 1, P =2
(7) MaxPooling	s = 2, k =2
(8) Dropout	r =0.2
(9) **Convolutionlayer5**	64 channels, k = 4, s = 1, P = 2*(with Batchnormalization)*
(10) Convolutionlayer6	64 channels, k = 3, s = 1, P =0
(11) Dropout	r = 0.4
(12) **Convolutionlayer7**	64 channels, k = 3, s = 1, P = 0*(with Batchnormalization)*
(13) **Convolutionlayer8**	64 channels, k = 3, s = 1, P = 0*(with Batchnormalization)*
(14) Dropout	r =0.5
(15) Fully connectedlayer1	500units
(16) Fully connectedlayer2	500units
(17) Softmaxfunction	

(k: kernal_size; s: stride; P: padding; r: dropping rate)

1.2 Optimizationmethod

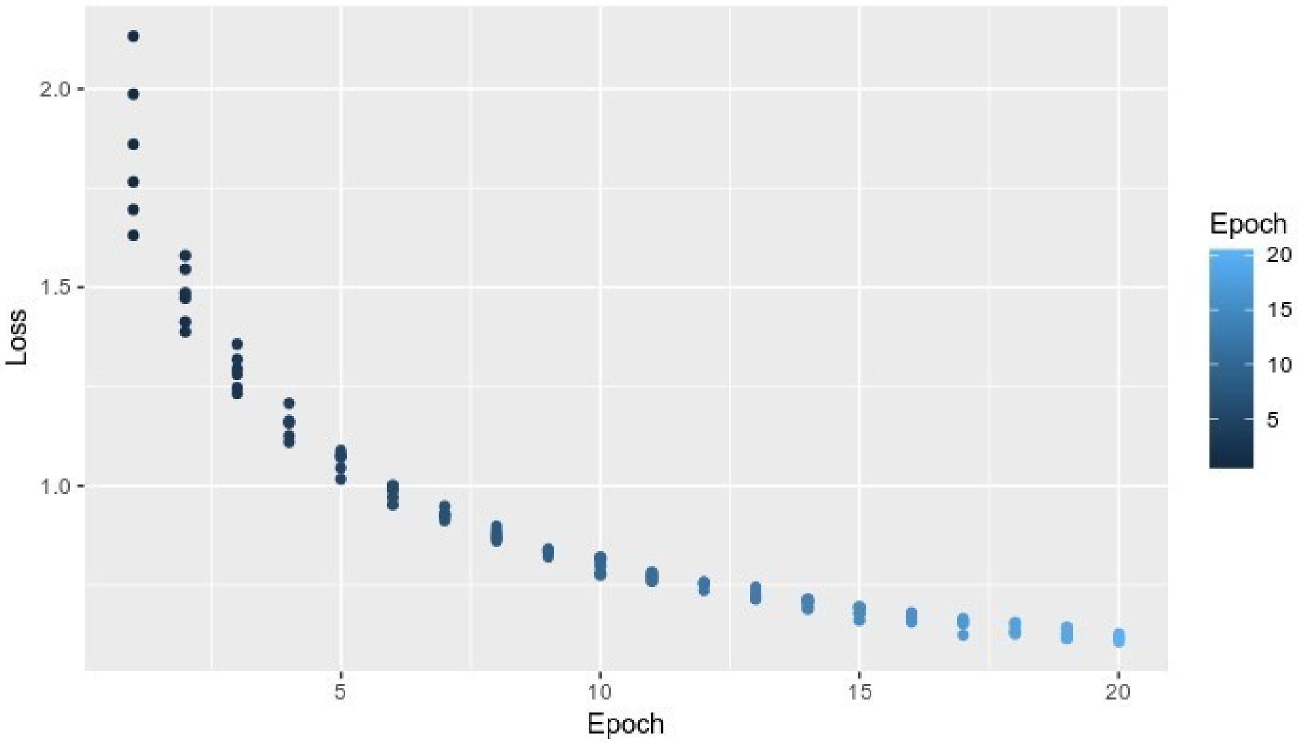
We use**ADAM**for optimization method with**learning rate of 0.001**. Note that we also apply data augmentation on training set composed by**random crop**and**horizontal flip**.

2. Results

2.1 Training

The following paragraph describe the loss during training epoch. We compute the loss for every 2000 steps in each epoch bytorch.nn.CrossEntropyLoss().

Finally, our model gets**82% accuracy**on test set.


Loss
Epoch
Epoch