HW2_RNN_Positive_Reviews

October 23, 2023

0.1 Homework 2: RNN

Problem 1: Text Generation (fake reviews) Text generation serves various purposes like machine translation, chatbots, virtual assistants, and AIGC. To understand the text generation process with a language model, consider it as an iterative approach. Initially, we predict the first word from the input sequence and subsequently use that predicted word in the input to generate the second word. Repeat the process until done.

If we recall, we focused on letter generation in class. Now, using the review data in HW1, let's build upon the in-class examples to create a simulated review using a GRU or LSTM model. For simplicity's sake, let's concentrate on coding the training process without delving into hyperparameter tuning or model evaluations. We'll generate both a negative review (combining 1-star, 2-star, and 3-star comments) and a positive review (utilizing 4-star and 5-star comments). Essentially, our goal is to generate tokens instead of individual letters.

1 Positive reviews next word prediction

[1]:
```
# import google drive

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

1.1 Read the Amazon comments data

[2]:
```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
import string


import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
```

[11]:
```
# import nltk
# nltk.download('all')
```

[12]:
```
os.chdir('/content/drive/MyDrive/Text Analytics/HW2')
```

[14]:
```
df = pd.read_csv('Amazon_Comments.csv',  delimiter="^", header=None,␣
 ↪names=["No", "Title", "Date", "Bool", "Review", "Rating"])

df = df.reset_index(drop=True)

df.set_index('No', inplace=True)

df.head()
```

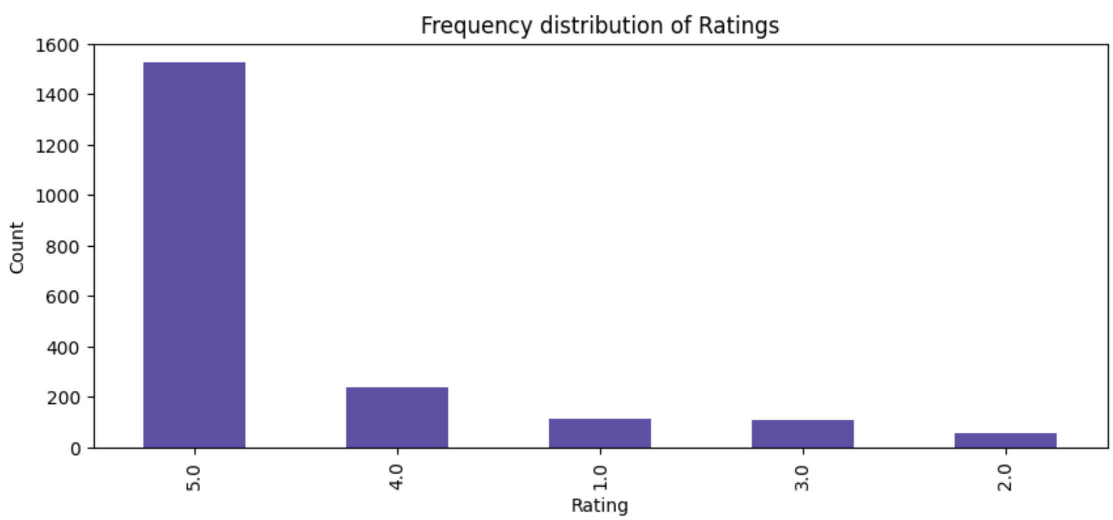
[14]:
```
                                                Title        Date   Bool  \
No
1   These are hands down the best quality bands fo…  2016-01-16  False
2                                  High Quality Bands  2016-01-22  False
3                                          Five Stars  2015-12-27  False
4   The resistance is great. I would agree that th…  2016-01-13  False
5                                Good quality product  2016-01-20  False

                                               Review  Rating
No
1    These are hands down the best quality bands f…     5.0
2    I just got this set yesterday as well as a se…     5.0
3    My husband uses these and finds them to be go…     5.0
4    I got these for Christmas and have been using…     4.0
5    Haven\t had it long enough to use all of the …     5.0
```

[15]:
```
# Drop Title, date, bool from df

df1 = df.copy()

df1.drop(columns=['Title', 'Date', 'Bool'], inplace=True)

df1.head()
```

[15]:
```
                                              Review  Rating
No
1     These are hands down the best quality bands f…     5.0
2     I just got this set yesterday as well as a se…     5.0
3     My husband uses these and finds them to be go…     5.0
4     I got these for Christmas and have been using…     4.0
5     Haven\t had it long enough to use all of the …     5.0
```

1.2 Visualize the frequency distribution of Ratings

[16]:
```python
# Visualize the frequency distribution of Ratings

df1['Rating'].value_counts().plot(kind='bar', colormap="Spectral_r",␣
 ↪figsize=(10,4))
plt.xlabel('Rating')
plt.ylabel('Count')
plt.title('Frequency distribution of Ratings')
plt.show()
```



The highest number of ratings in the Amazon review data consist of 5 star ratings followed by 4 star and 1 star ratings respectively.

1.3 Label the data as positive and negative

[17]:
```python
# Add sentiment column -- If Rating =1 or 2 or 3 then label that review as␣
 ↪negative else Flag as Positive

df1['sentiment'] = df['Rating'].map({1.0: 'Negative', 2.0: 'Negative', 3.0:␣
 ↪'Negative', 4.0: 'Positive', 5.0: 'Positive'})
```

```python
df1
```

[17]:

```
                                                Review  Rating sentiment
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                  …     …         …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

[2038 rows x 3 columns]
```

[18]:

```python
# Label the positive sentiment as 1 and negative sentiment as 0

df1['Label'] = df1['sentiment'].map({'Positive': 1, 'Negative': 0})
df1
```

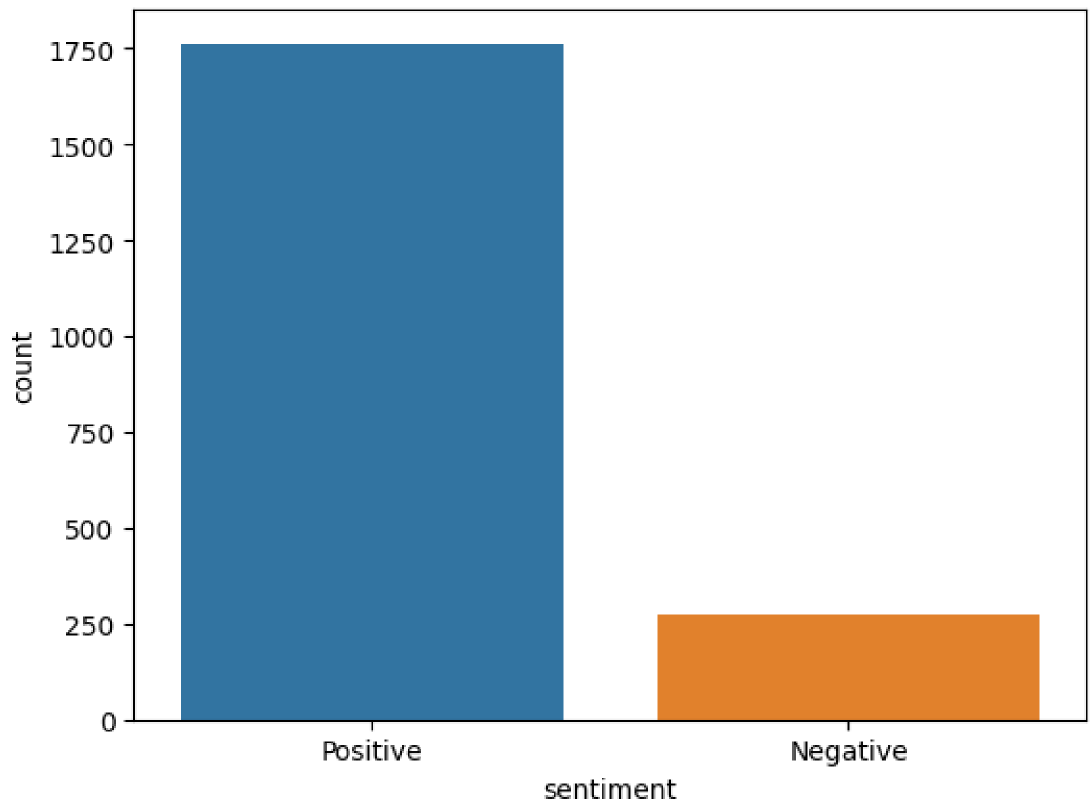
[18]:

```
                                                Review  Rating sentiment  \
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                  …     …         …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

      Label
No
1         1
2         1
3         1
4         1
5         1
…         …
2034      1
2035      1
```

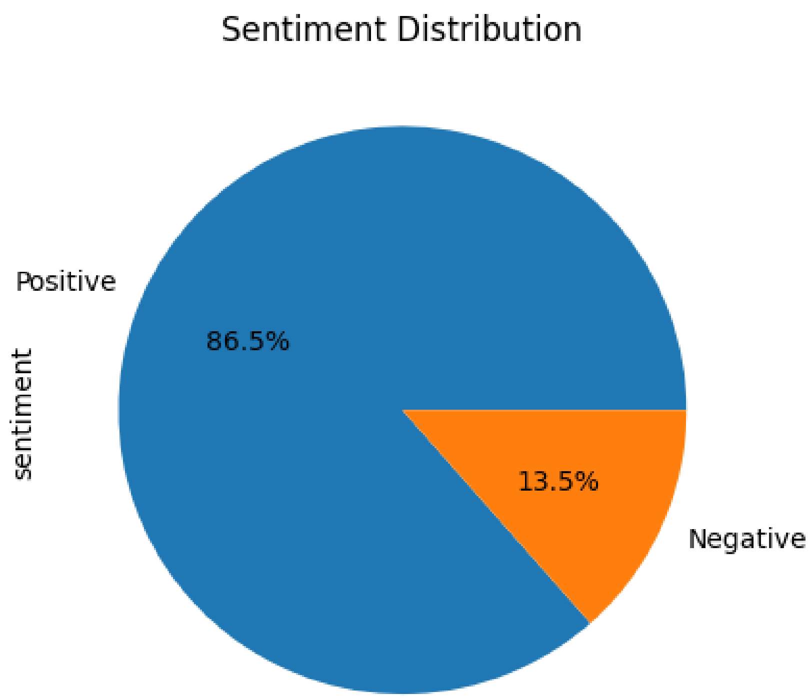
```
2036      1
2037      1
2038      0

[2038 rows x 4 columns]
```

1.4 Visualize the distribution of the Sentiment

[19]:
```python
# Visualize a count plot for the sentiment column with sns.countplot

import seaborn as sns
sns.countplot(x='sentiment', data=df1)
plt.show()
```



[20]:
```python
# Visualize a pie chart for the sentiment column

df1.sentiment.value_counts().plot.pie(autopct='%1.1f%%')
plt.title('Sentiment Distribution')
plt.show()
```



1.5 Data Cleaning

[21]:
```
# Clean the review data

# Function to clean and preprocess text
def clean_text(text):

    # Remove special characters and numbers
    text = re.sub(r'[^a-zA-Z\s]', '', text)

    # Convert text to lowercase
    text = text.lower()

    # Remove punctuation
    text = ''.join([char for char in text if char not in string.punctuation])


    # Tokenize text
    tokens = nltk.word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
```

```
    tokens = [word for word in tokens if word not in stop_words and not word.
 ↪isdigit()]

    #Stemming (you can replace with lemmatization if preferred)
    #stemmer = PorterStemmer()

    #tokens = [stemmer.stem(word) for word in tokens]

    # Create a lemmatizer object.
    lemmatizer = WordNetLemmatizer()

   #Lemmatization
    tokens = [lemmatizer.lemmatize(word) for word in tokens]

    # Reconstruct cleaned text
    cleaned_text = ' '.join(tokens)

    return cleaned_text
```

[22]:
```
# Apply the clean_text function to each review in the DataFrame

df2 = df1.copy()

df2['Clean_Review'] = df1['Review'].apply(clean_text)

# Print the cleaned reviews

df2.head()
```

[22]:
```
                                               Review  Rating sentiment  \
No
1    These are hands down the best quality bands f…     5.0  Positive
2    I just got this set yesterday as well as a se…     5.0  Positive
3    My husband uses these and finds them to be go…     5.0  Positive
4    I got these for Christmas and have been using…     4.0  Positive
5    Haven\t had it long enough to use all of the …     5.0  Positive

    Label                                      Clean_Review
No
1       1  hand best quality band money year old male wan…
2       1  got set yesterday well set another company cou…
3       1  husband us find good arthritis stretching exer…
4       1  got christmas using multiple day week since re…
5       1  havent long enough use component far im impres…
```

[23]:
```
# Generate tokens for Clean_Review

df2['Clean_review_Tokens'] = df2['Clean_Review'].apply(nltk.word_tokenize)

df2
```

[23]:
```
                                                Review  Rating sentiment  \
No
1     These are hands down the best quality bands f…     5.0  Positive
2     I just got this set yesterday as well as a se…     5.0  Positive
3     My husband uses these and finds them to be go…     5.0  Positive
4     I got these for Christmas and have been using…     4.0  Positive
5     Haven\t had it long enough to use all of the …     5.0  Positive
…                                                  …     …       …
2034                            Just 1 like Nonna\s!     5.0  Positive
2035                                    Works great!     5.0  Positive
2036                                       very good     5.0  Positive
2037                                           great     5.0  Positive
2038  material changed and effect so easy. couple u…     1.0  Negative

      Label                                       Clean_Review  \
No
1         1  hand best quality band money year old male wan…
2         1  got set yesterday well set another company cou…
3         1  husband us find good arthritis stretching exer…
4         1  got christmas using multiple day week since re…
5         1  havent long enough use component far im impres…
…       …                                                  …
2034      1                                       1 like nonnas
2035      1                                          work great
2036      1                                                good
2037      1                                               great
2038      0  material changed effect easy couple usage alum…

                                    Clean_review_Tokens
No
1     [hand, best, quality, band, money, year, old, …
2     [got, set, yesterday, well, set, another, comp…
3     [husband, us, find, good, arthritis, stretchin…
4     [got, christmas, using, multiple, day, week, s…
5     [havent, long, enough, use, component, far, im…
…                                                  …
2034                                  [1, like, nonnas]
2035                                      [work, great]
2036                                             [good]
2037                                            [great]
2038  [material, changed, effect, easy, couple, usag…
```

```
[2038 rows x 6 columns]
```

1.6 Filter the Positive Reviews Data

[24]:
```python
# Subset the positive sentiment data

positive_df = df2[df2['sentiment'] == 'Positive']
positive_df
```

[24]:
```
                                                Review  Rating sentiment  \
No
1       These are hands down the best quality bands f…     5.0  Positive
2       I just got this set yesterday as well as a se…     5.0  Positive
3       My husband uses these and finds them to be go…     5.0  Positive
4       I got these for Christmas and have been using…     4.0  Positive
5       Haven\t had it long enough to use all of the …     5.0  Positive
…                                                    …     …       …
2033                               Best espresso maker!     5.0  Positive
2034                               Just 1 like Nonna\s!     5.0  Positive
2035                                       Works great!     5.0  Positive
2036                                          very good     5.0  Positive
2037                                              great     5.0  Positive

      Label                                    Clean_Review  \
No
1         1  hand best quality band money year old male wan…
2         1  got set yesterday well set another company cou…
3         1  husband us find good arthritis stretching exer…
4         1  got christmas using multiple day week since re…
5         1  havent long enough use component far im impres…
…       …                                                …
2033      1                              best espresso maker
2034      1                                     1 like nonnas
2035      1                                        work great
2036      1                                              good
2037      1                                             great

                                   Clean_review_Tokens
No
1      [hand, best, quality, band, money, year, old, …
2      [got, set, yesterday, well, set, another, comp…
3      [husband, us, find, good, arthritis, stretchin…
4      [got, christmas, using, multiple, day, week, s…
5      [havent, long, enough, use, component, far, im…
…                                                    …
2033                             [best, espresso, maker]
```

```
2034                                   [l, like, nonnas]
2035                                       [work, great]
2036                                              [good]
2037                                             [great]

[1763 rows x 6 columns]
```

1.7 Unigram Token Frequency for Positive Data

[25]:
```python
import collections
from collections import Counter
from itertools import chain

word_tokenize = nltk.word_tokenize

# Tokenize the text column

# Print the tokenized text
corpus = positive_df['Clean_review_Tokens']
corpus = corpus.tolist()
# Flatten list of lists to a single list
tokens = list(chain(*corpus))
unique_freq = collections.Counter(tokens)
# Count each unique element
unique_freq_df = pd.DataFrame.from_dict(unique_freq, orient='index').
 ↪reset_index() # Convert to dataframe
# Rename columns
unique_freq_df = unique_freq_df.rename(columns={'index': 'Token', 0: 'Count'})
# Sort by count
unique_freq_df.sort_values('Count', ascending=False, inplace=True)
unique_freq_df = unique_freq_df

unique_freq_df1 = unique_freq_df.reset_index(drop=True)
unique_freq_df2 = unique_freq_df1.set_index("Token")
print(len(unique_freq_df))
unique_freq_df2
```

```
4042
```

[25]:
```
         Count
Token
great      594
work       363
use        324
love       315
product    291
…          …
```

```
loss         1
meltit       1
alone        1
methe        1
nonnas       1

[4042 rows x 1 columns]
```

[26]:

```
# Storing the frequencies of unigrams

freq1 = unique_freq_df['Count'].reset_index(drop=True)
print(type(freq1))
freq1
```

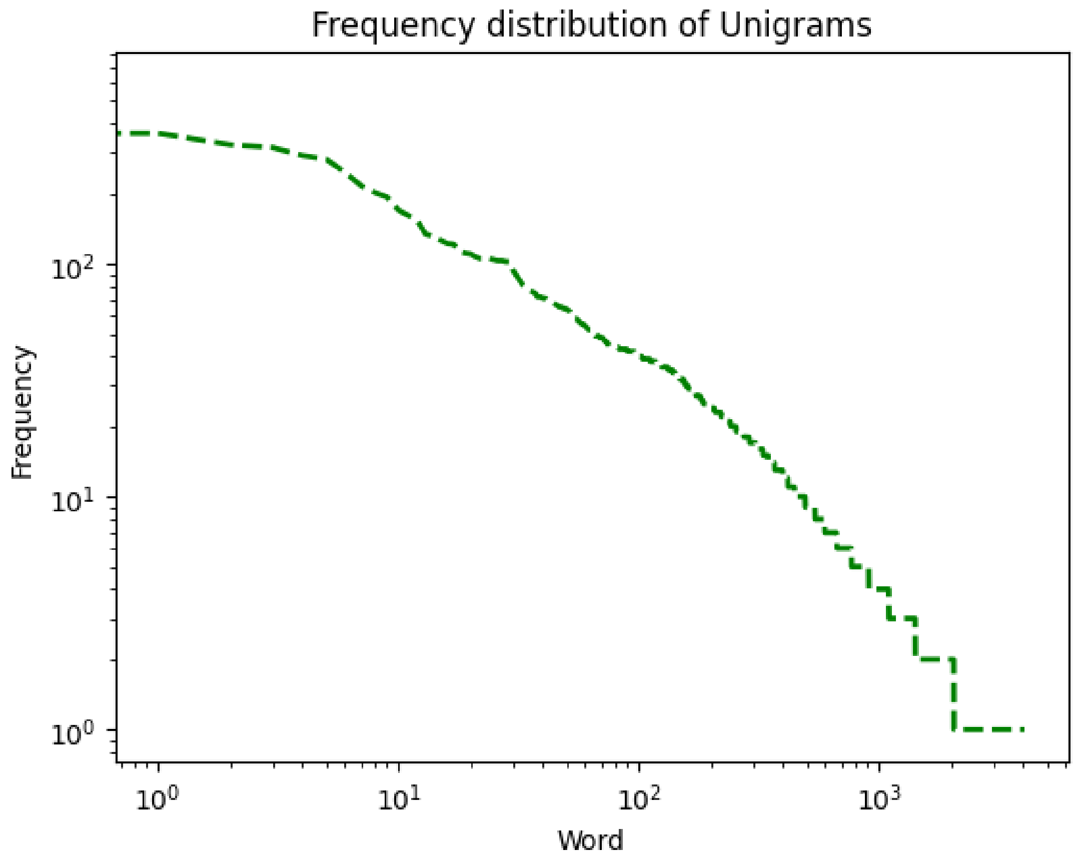
```
<class 'pandas.core.series.Series'>
```

[26]:

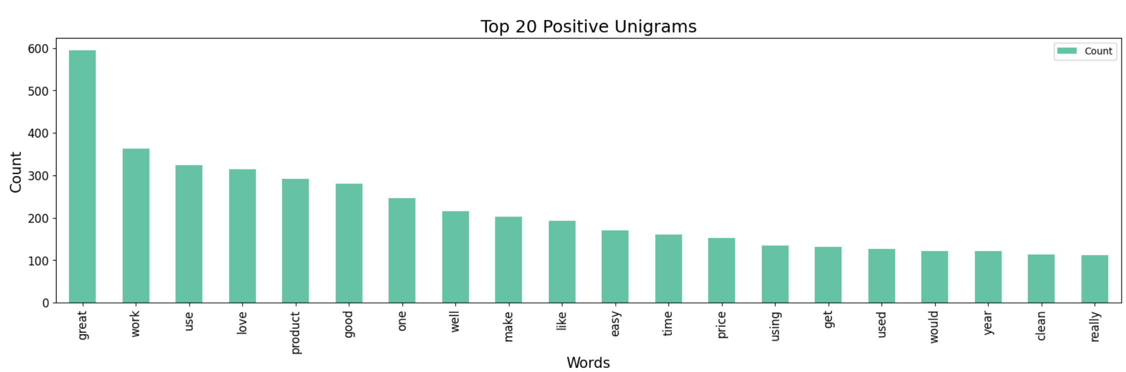
```
0       594
1       363
2       324
3       315
4       291
       …
4037      1
4038      1
4039      1
4040      1
4041      1
Name: Count, Length: 4042, dtype: int64
```

[27]:

```
# Plotting the log scale of frequencies of unigrams

plt.plot(freq1, color="Green", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Unigrams")
plt.show()
```



[28]:
```python
# Plot the df3 Unigrams

unique_freq_df2.head(20).plot(kind='bar', figsize=(20, 5), cmap="Set2")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Positive Unigrams', fontsize=18)
plt.show()
```



1.8 Bigram Token Frequency for Positive Data

[29]:
```
# Generate bigrams from df2['Clean_review_Tokens']

bigram_list = [list(nltk.bigrams(text)) for text in␣
↪positive_df['Clean_review_Tokens']]

# Create a Counter object to count the frequency of each bigram
bigram_count = collections.Counter(list(chain(*bigram_list)))

# Convert the Counter object to a DataFrame
bigram_df = pd.DataFrame.from_dict(bigram_count, orient='index').reset_index()

# Rename the columns
bigram_df = bigram_df.rename(columns={'index': 'Bigram', 0: 'Count'})

# Sort the DataFrame by frequency in descending order
bigram_df.sort_values('Count', ascending=False, inplace=True)

# Print the top 20 bigrams
bigram_df.head(20)

bigram_df1 = bigram_df.reset_index(drop=True)
bigram_df2 = bigram_df1.set_index("Bigram")
bigram_df2
```

[29]:
```
                      Count
Bigram
(work, great)           121
(work, well)             68
(great, product)         66
(easy, use)              53
(great, price)           40
…                      …
(heavily, handle)         1
(handle, adjustment)      1
(adjustment, lever)       1
(lever, appear)           1
(like, nonnas)            1

[18938 rows x 1 columns]
```

[30]:
```
# Storing the frequencies of unigrams
```

```python
freq2 = bigram_df2['Count'].reset_index(drop=True)
print(type(freq2))
freq2
```

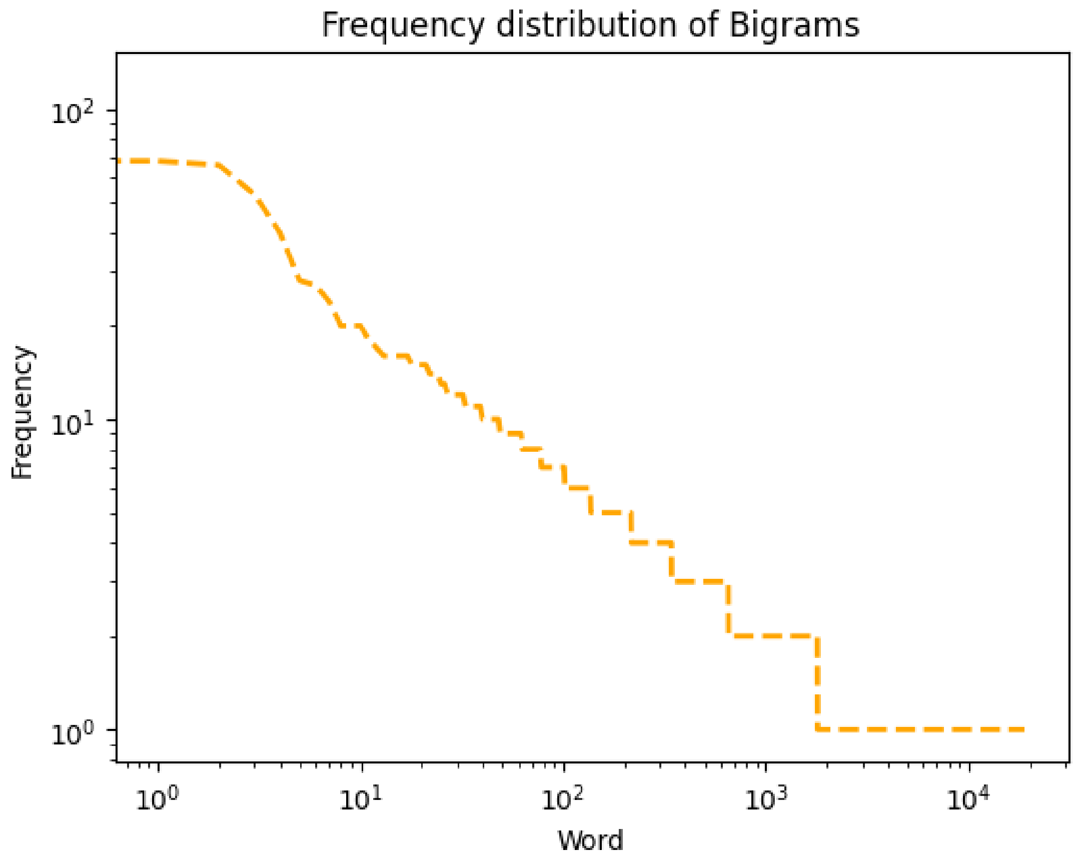
```
<class 'pandas.core.series.Series'>
```

[30]:

```
0        121
1         68
2         66
3         53
4         40
        …
18933      1
18934      1
18935      1
18936      1
18937      1
Name: Count, Length: 18938, dtype: int64
```

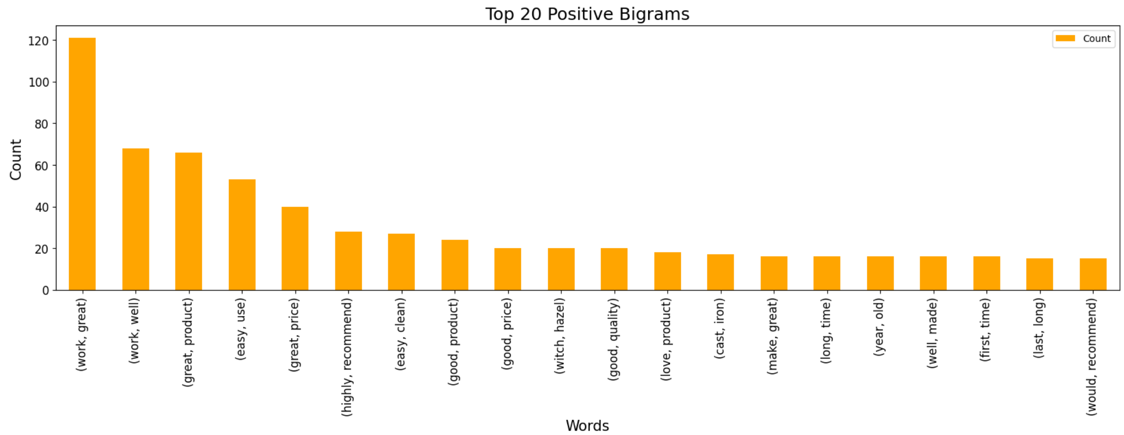
[31]:

```python
# Plotting the log scale of frequencies of Bigrams

plt.plot(freq2, color="Orange", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Bigrams")
plt.show()
```



[32]:

```
# Plot the df3 Bigrams frequencies

bigram_df2.head(20).plot(kind='bar', figsize=(20, 5), color="Orange")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Positive Bigrams', fontsize=18)
plt.show()
```



1.9 Trigram Frequency

[33]:
```
# Generate trigrams from df2['Clean_review_Tokens']

trigram_list = [list(nltk.trigrams(text)) for text in␣
    ↪positive_df['Clean_review_Tokens']]

# Create a Counter object to count the frequency of each trigram
trigram_count = collections.Counter(list(chain(*trigram_list)))

# Convert the Counter object to a DataFrame
trigram_df = pd.DataFrame.from_dict(trigram_count, orient='index').reset_index()

# Rename the columns
trigram_df = trigram_df.rename(columns={'index': 'Trigram', 0: 'Count'})

# Sort the DataFrame by frequency in descending order
trigram_df.sort_values('Count', ascending=False, inplace=True)

# Print the top 20 trigrams
trigram_df.head(20)

trigram_df1 = trigram_df.reset_index(drop=True)
trigram_df2 = trigram_df1.set_index("Trigram")
trigram_df2
```

[33]:
```
                          Count
Trigram
(last, long, time)            8
(great, product, great)       6
(cast, iron, pan)             6
```

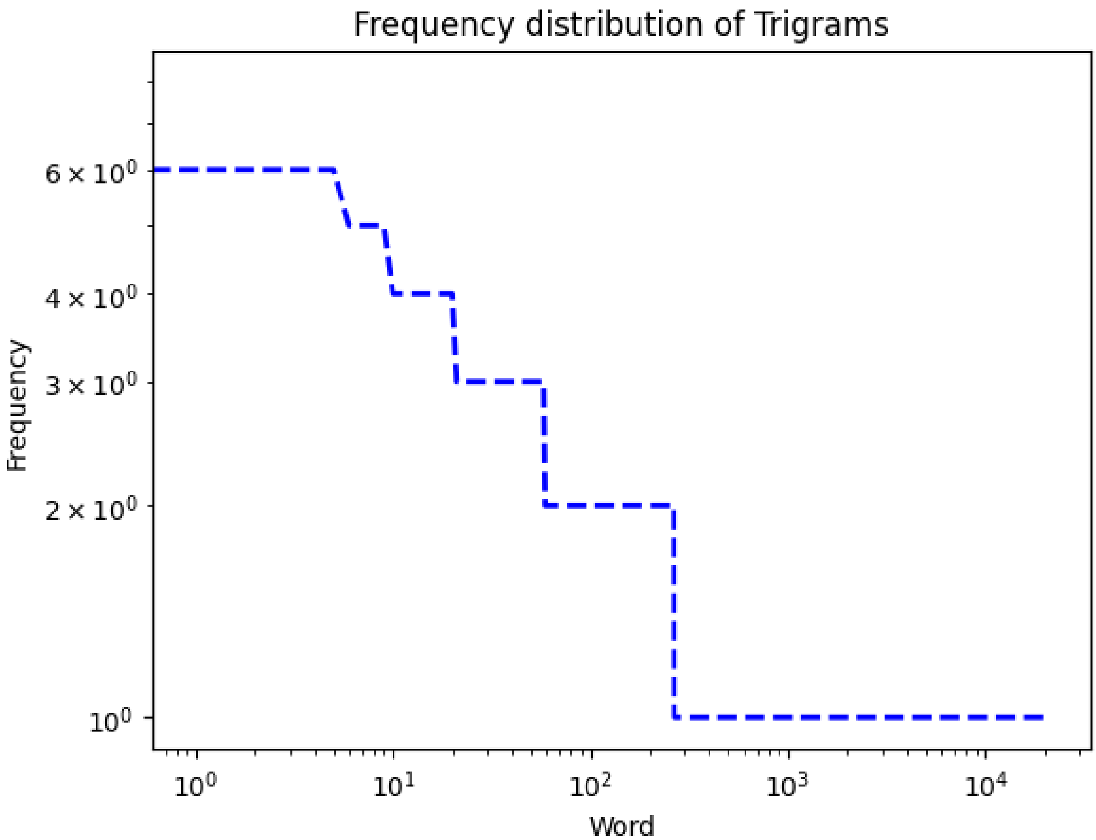
```
(work, really, well)          6
(work, great, love)           6
…                           …
(brand, neither, worked)      1
(two, brand, neither)         1
(bought, two, brand)          1
(great, bought, two)          1
(l, like, nonnas)             1

[20878 rows x 1 columns]
```

[34]:
```python
# Storing the frequencies of unigrams

freq3 = trigram_df2['Count'].reset_index(drop=True)
print(type(freq3))
freq3
```

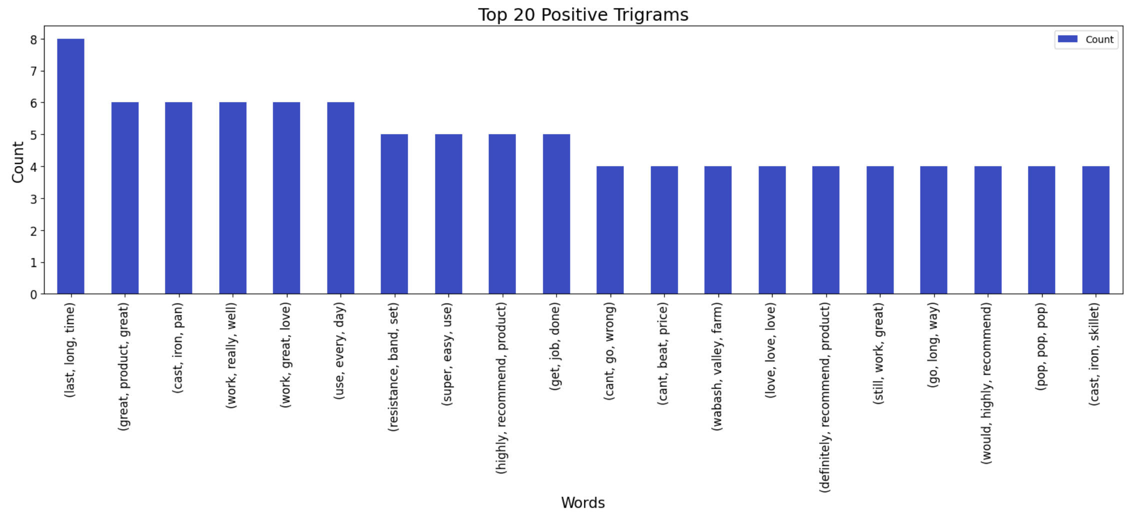
```
<class 'pandas.core.series.Series'>
```

[34]:
```
0        8
1        6
2        6
3        6
4        6
        ..
20873    1
20874    1
20875    1
20876    1
20877    1
Name: Count, Length: 20878, dtype: int64
```

[35]:
```python
# Plotting the log scale of frequencies of unigrams

plt.plot(freq3, color="Blue", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Trigrams")
plt.show()
```

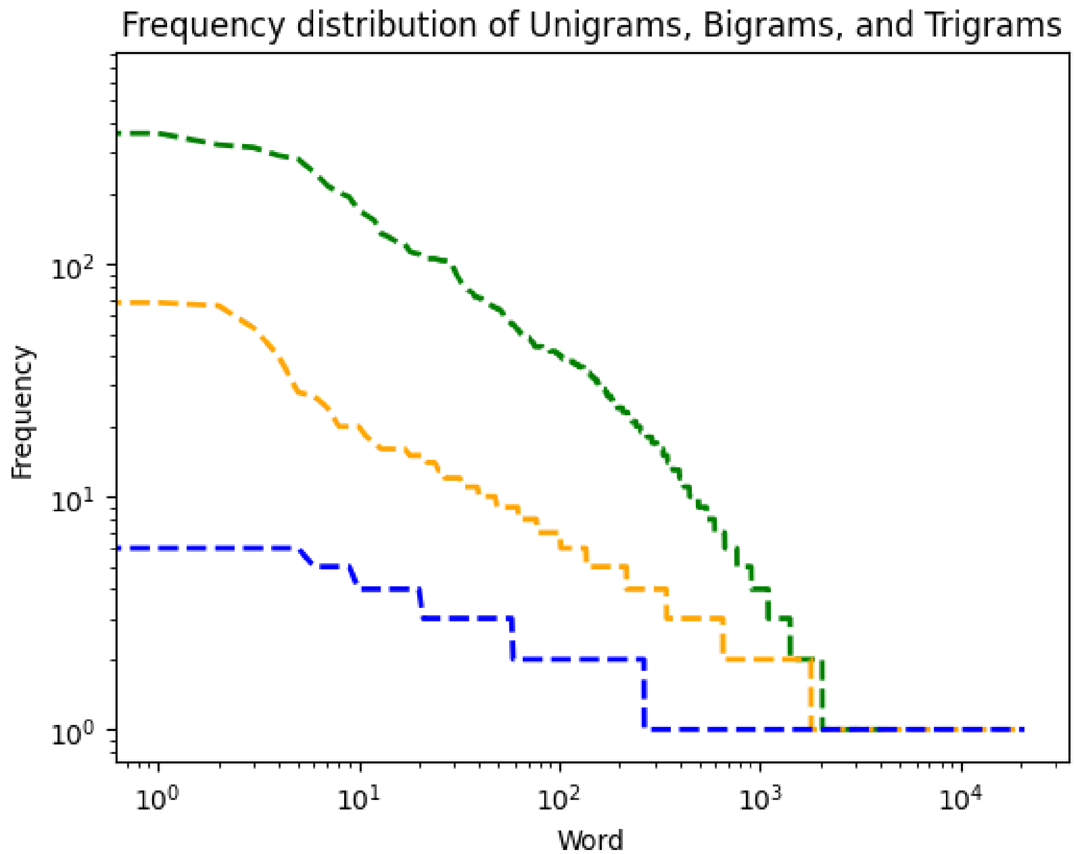


[36]:
```
# Plot the top 20 Trigrams

trigram_df2.head(20).plot(kind='bar', figsize=(20, 5), cmap="coolwarm")
plt.xlabel('Words', fontsize = 15)
plt.ylabel('Count', fontsize=15)
plt.xticks(size = 12)
plt.yticks(size = 12)
plt.title('Top 20 Positive Trigrams', fontsize=18)
plt.show()
```



1.10 Compare the Unigram, Bigram and Trigram token counts distributions

[37]:
```
# Compare the Frequency distributions of Unigrams, Bigrams, and Trigrams

plt.plot(freq1, color="Green", linewidth=2, linestyle='--')
plt.plot(freq2, color="Orange", linewidth=2, linestyle='--')
plt.plot(freq3, color="Blue", linewidth=2, linestyle='--')
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Word")
plt.ylabel("Frequency")
plt.title("Frequency distribution of Unigrams, Bigrams, and Trigrams")
plt.show()
```



The graph shows the frequency distribution of token counts in unigrams, bigrams, and trigrams in the positive sentiment reviews. The x-axis represents the words and the y-axis represents the frequency of occurrence. The graph shows that unigrams are the most frequent type of token, followed by bigrams and trigrams. This is because unigrams are the smallest units of meaning, and they are more likely to occur in a sentence than bigrams or trigrams. The graph also shows that there is a long tail of low-frequency tokens, which means that there are many words that only occur once or twice in the reviews. This is not surprising, as there are a large number of possible words that can be used to express a sentiment.

1.11 Next word prediction using LSTM

[38]:
```
#Import the necessary libraries

import tensorflow as tf
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.layers import BatchNormalization
from keras.layers import Dropout
```

1.12 Data Pre-processing

[39]:
```
# Tokenize the text data

tokenizer = Tokenizer()
tokenizer.fit_on_texts(positive_df['Clean_review_Tokens'])
total_words = len(tokenizer.word_index) + 1
print(total_words)
```

```
4043
```

[40]:
```
# Create input sequences
input_sequences = []
for line in positive_df['Clean_review_Tokens']:
    token_list = tokenizer.texts_to_sequences([line])[0]

    for i in range(1, len(token_list)):
        n_gram_sequence = token_list[:i+1]
        input_sequences.append(n_gram_sequence)
```

[41]:
```
# View input_sequences

# input_sequences
```

[42]:
```
# Print the max sequence length

max_sequence_len = max([len(seq) for seq in input_sequences])
print(max_sequence_len)
```

```
195
```

[43]:
```
# Pad sequences and split into predictors and label

padded_input_sequences = np.array(pad_sequences(input_sequences,␣
↪maxlen=max_sequence_len, padding='pre'))

print(padded_input_sequences)
```

```
[[   0    0    0 …    0   67   29]
 [   0    0    0 …   67   29   21]
 [   0    0    0 …   29   21   98]
 …
```

```
 [    0     0     0 …     0 4041    10]
 [    0     0     0 … 4041    10  4042]
 [    0     0     0 …     0     2     1]]
```

[44]:
```python
# Define X and y

X = padded_input_sequences[:, :-1]
y = padded_input_sequences[:, -1]

# Print the dimensions of X and y

print("The shape of padded input sequence X is:", X.shape)
print("The shape of padded input sequence y is:", y.shape)
```

```
The shape of padded input sequence X is: (22859, 194)
The shape of padded input sequence y is: (22859,)
```

[45]:
```python
# Convert target data to one-hot encoding
y = tf.keras.utils.to_categorical(y, num_classes=total_words)

# Print the new dimensions of X and y
print("The shape of X is:", X.shape)
print("The shape of y is:", y.shape)
```

```
The shape of X is: (22859, 194)
The shape of y is: (22859, 4043)
```

[46]:
```python
# Define the model

model = Sequential()
model.add(Embedding(total_words, 100, input_length=max_sequence_len-1))
model.add(LSTM(128, return_sequences=True))
model.add((LSTM(128)))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.01),
 ↪metrics=['accuracy'])
model.summary()
```

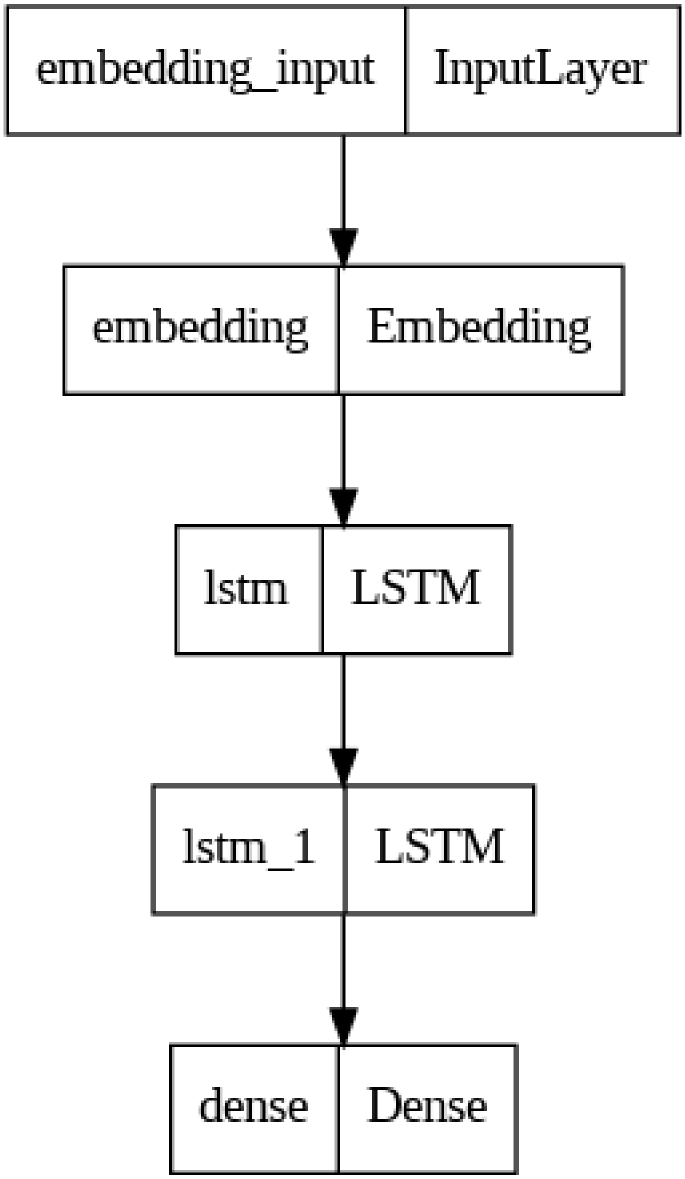
```
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate`
or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.Adam.

Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 194, 100)          404300


 lstm (LSTM)                 (None, 194, 128)          117248
```

```
 lstm_1 (LSTM)               (None, 128)               131584

 dense (Dense)               (None, 4043)              521547

=================================================================
Total params: 1174679 (4.48 MB)
Trainable params: 1174679 (4.48 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[47]:

```python
# Visualize model configuration

from tensorflow import keras
from tensorflow.keras.utils import plot_model

keras.utils.plot_model(model, to_file='model.png', show_layer_names=True)
```

[47]:



[48]:

```
#Fit the model

history = model.fit(X, y, validation_split=0.05, batch_size=128, epochs=500,
    shuffle=True).history
```

```
Epoch 1/500
170/170 [==============================] - 39s 182ms/step - loss: 7.4800 -
accuracy: 0.0158 - val_loss: 7.4009 - val_accuracy: 0.0105
Epoch 2/500
170/170 [==============================] - 18s 105ms/step - loss: 7.1589 -
accuracy: 0.0170 - val_loss: 7.4830 - val_accuracy: 0.0271
```

```
Epoch 3/500
170/170 [==============================] - 13s 79ms/step - loss: 7.0930 -
accuracy: 0.0181 - val_loss: 7.5439 - val_accuracy: 0.0271
Epoch 4/500
170/170 [==============================] - 11s 63ms/step - loss: 7.0436 -
accuracy: 0.0192 - val_loss: 7.6035 - val_accuracy: 0.0236
Epoch 5/500
170/170 [==============================] - 9s 51ms/step - loss: 6.9798 -
accuracy: 0.0208 - val_loss: 7.6391 - val_accuracy: 0.0271
Epoch 6/500
170/170 [==============================] - 8s 44ms/step - loss: 6.9053 -
accuracy: 0.0223 - val_loss: 7.6959 - val_accuracy: 0.0262
Epoch 7/500
170/170 [==============================] - 6s 38ms/step - loss: 6.8268 -
accuracy: 0.0246 - val_loss: 7.7662 - val_accuracy: 0.0262
Epoch 8/500
170/170 [==============================] - 7s 41ms/step - loss: 6.7491 -
accuracy: 0.0268 - val_loss: 7.7954 - val_accuracy: 0.0280
Epoch 9/500
170/170 [==============================] - 6s 36ms/step - loss: 6.6699 -
accuracy: 0.0293 - val_loss: 7.8160 - val_accuracy: 0.0254
Epoch 10/500
170/170 [==============================] - 7s 41ms/step - loss: 6.5894 -
accuracy: 0.0309 - val_loss: 7.8773 - val_accuracy: 0.0254
Epoch 11/500
170/170 [==============================] - 7s 40ms/step - loss: 6.5083 -
accuracy: 0.0332 - val_loss: 7.9550 - val_accuracy: 0.0280
Epoch 12/500
170/170 [==============================] - 6s 34ms/step - loss: 6.4254 -
accuracy: 0.0341 - val_loss: 8.0108 - val_accuracy: 0.0245
Epoch 13/500
170/170 [==============================] - 5s 31ms/step - loss: 6.3438 -
accuracy: 0.0364 - val_loss: 8.0925 - val_accuracy: 0.0254
Epoch 14/500
170/170 [==============================] - 6s 35ms/step - loss: 6.2606 -
accuracy: 0.0397 - val_loss: 8.1254 - val_accuracy: 0.0289
Epoch 15/500
170/170 [==============================] - 6s 36ms/step - loss: 6.1759 -
accuracy: 0.0413 - val_loss: 8.1943 - val_accuracy: 0.0271
Epoch 16/500
170/170 [==============================] - 5s 31ms/step - loss: 6.0981 -
accuracy: 0.0421 - val_loss: 8.2738 - val_accuracy: 0.0289
Epoch 17/500
170/170 [==============================] - 5s 30ms/step - loss: 6.0131 -
accuracy: 0.0468 - val_loss: 8.3426 - val_accuracy: 0.0297
Epoch 18/500
170/170 [==============================] - 6s 33ms/step - loss: 5.9315 -
accuracy: 0.0486 - val_loss: 8.4109 - val_accuracy: 0.0306
```

```
Epoch 19/500
170/170 [==============================] - 5s 32ms/step - loss: 5.8488 -
accuracy: 0.0516 - val_loss: 8.4704 - val_accuracy: 0.0297
Epoch 20/500
170/170 [==============================] - 5s 31ms/step - loss: 5.7638 -
accuracy: 0.0547 - val_loss: 8.5336 - val_accuracy: 0.0324
Epoch 21/500
170/170 [==============================] - 5s 31ms/step - loss: 5.6788 -
accuracy: 0.0595 - val_loss: 8.6105 - val_accuracy: 0.0315
Epoch 22/500
170/170 [==============================] - 5s 31ms/step - loss: 5.5944 -
accuracy: 0.0619 - val_loss: 8.7007 - val_accuracy: 0.0324
Epoch 23/500
170/170 [==============================] - 5s 32ms/step - loss: 5.5116 -
accuracy: 0.0669 - val_loss: 8.7303 - val_accuracy: 0.0306
Epoch 24/500
170/170 [==============================] - 5s 29ms/step - loss: 5.4286 -
accuracy: 0.0712 - val_loss: 8.7951 - val_accuracy: 0.0341
Epoch 25/500
170/170 [==============================] - 6s 35ms/step - loss: 5.3425 -
accuracy: 0.0775 - val_loss: 8.8801 - val_accuracy: 0.0341
Epoch 26/500
170/170 [==============================] - 6s 34ms/step - loss: 5.2586 -
accuracy: 0.0858 - val_loss: 8.9572 - val_accuracy: 0.0324
Epoch 27/500
170/170 [==============================] - 5s 32ms/step - loss: 5.1752 -
accuracy: 0.0936 - val_loss: 9.0003 - val_accuracy: 0.0324
Epoch 28/500
170/170 [==============================] - 6s 33ms/step - loss: 5.0926 -
accuracy: 0.0993 - val_loss: 9.0861 - val_accuracy: 0.0332
Epoch 29/500
170/170 [==============================] - 6s 34ms/step - loss: 5.0089 -
accuracy: 0.1096 - val_loss: 9.1545 - val_accuracy: 0.0315
Epoch 30/500
170/170 [==============================] - 6s 33ms/step - loss: 4.9259 -
accuracy: 0.1209 - val_loss: 9.2317 - val_accuracy: 0.0350
Epoch 31/500
170/170 [==============================] - 6s 33ms/step - loss: 4.8436 -
accuracy: 0.1302 - val_loss: 9.2825 - val_accuracy: 0.0306
Epoch 32/500
170/170 [==============================] - 6s 33ms/step - loss: 4.7620 -
accuracy: 0.1414 - val_loss: 9.3955 - val_accuracy: 0.0324
Epoch 33/500
170/170 [==============================] - 5s 31ms/step - loss: 4.6791 -
accuracy: 0.1573 - val_loss: 9.4622 - val_accuracy: 0.0262
Epoch 34/500
170/170 [==============================] - 5s 31ms/step - loss: 4.6024 -
accuracy: 0.1672 - val_loss: 9.5402 - val_accuracy: 0.0332
```

```
Epoch 35/500
170/170 [==============================] - 5s 31ms/step - loss: 4.5206 -
accuracy: 0.1820 - val_loss: 9.6235 - val_accuracy: 0.0341
Epoch 36/500
170/170 [==============================] - 5s 32ms/step - loss: 4.4414 -
accuracy: 0.1979 - val_loss: 9.6711 - val_accuracy: 0.0324
Epoch 37/500
170/170 [==============================] - 5s 30ms/step - loss: 4.3633 -
accuracy: 0.2115 - val_loss: 9.7440 - val_accuracy: 0.0289
Epoch 38/500
170/170 [==============================] - 5s 30ms/step - loss: 4.2926 -
accuracy: 0.2249 - val_loss: 9.8297 - val_accuracy: 0.0306
Epoch 39/500
170/170 [==============================] - 5s 31ms/step - loss: 4.2168 -
accuracy: 0.2356 - val_loss: 9.8818 - val_accuracy: 0.0297
Epoch 40/500
170/170 [==============================] - 5s 31ms/step - loss: 4.1395 -
accuracy: 0.2478 - val_loss: 9.9820 - val_accuracy: 0.0297
Epoch 41/500
170/170 [==============================] - 5s 31ms/step - loss: 4.0739 -
accuracy: 0.2573 - val_loss: 10.0668 - val_accuracy: 0.0289
Epoch 42/500
170/170 [==============================] - 5s 31ms/step - loss: 4.0022 -
accuracy: 0.2686 - val_loss: 10.1673 - val_accuracy: 0.0280
Epoch 43/500
170/170 [==============================] - 5s 32ms/step - loss: 3.9302 -
accuracy: 0.2818 - val_loss: 10.2262 - val_accuracy: 0.0332
Epoch 44/500
170/170 [==============================] - 5s 32ms/step - loss: 3.8618 -
accuracy: 0.2893 - val_loss: 10.3105 - val_accuracy: 0.0315
Epoch 45/500
170/170 [==============================] - 5s 30ms/step - loss: 3.7962 -
accuracy: 0.2997 - val_loss: 10.4454 - val_accuracy: 0.0280
Epoch 46/500
170/170 [==============================] - 5s 31ms/step - loss: 3.7356 -
accuracy: 0.3100 - val_loss: 10.4788 - val_accuracy: 0.0297
Epoch 47/500
170/170 [==============================] - 5s 31ms/step - loss: 3.6728 -
accuracy: 0.3173 - val_loss: 10.5773 - val_accuracy: 0.0315
Epoch 48/500
170/170 [==============================] - 5s 32ms/step - loss: 3.6121 -
accuracy: 0.3275 - val_loss: 10.6463 - val_accuracy: 0.0324
Epoch 49/500
170/170 [==============================] - 6s 33ms/step - loss: 3.5474 -
accuracy: 0.3378 - val_loss: 10.7515 - val_accuracy: 0.0324
Epoch 50/500
170/170 [==============================] - 5s 31ms/step - loss: 3.4919 -
accuracy: 0.3453 - val_loss: 10.8210 - val_accuracy: 0.0306
```

```
Epoch 51/500
170/170 [==============================] - 5s 32ms/step - loss: 3.4352 -
accuracy: 0.3549 - val_loss: 10.9504 - val_accuracy: 0.0324
Epoch 52/500
170/170 [==============================] - 5s 30ms/step - loss: 3.3784 -
accuracy: 0.3625 - val_loss: 11.0258 - val_accuracy: 0.0315
Epoch 53/500
170/170 [==============================] - 5s 29ms/step - loss: 3.3247 -
accuracy: 0.3699 - val_loss: 11.1067 - val_accuracy: 0.0324
Epoch 54/500
170/170 [==============================] - 5s 30ms/step - loss: 3.2726 -
accuracy: 0.3800 - val_loss: 11.2066 - val_accuracy: 0.0324
Epoch 55/500
170/170 [==============================] - 5s 29ms/step - loss: 3.2172 -
accuracy: 0.3870 - val_loss: 11.3060 - val_accuracy: 0.0332
Epoch 56/500
170/170 [==============================] - 5s 30ms/step - loss: 3.1675 -
accuracy: 0.3942 - val_loss: 11.3571 - val_accuracy: 0.0332
Epoch 57/500
170/170 [==============================] - 5s 32ms/step - loss: 3.1190 -
accuracy: 0.4028 - val_loss: 11.4296 - val_accuracy: 0.0350
Epoch 58/500
170/170 [==============================] - 5s 32ms/step - loss: 3.0732 -
accuracy: 0.4064 - val_loss: 11.5441 - val_accuracy: 0.0359
Epoch 59/500
170/170 [==============================] - 5s 31ms/step - loss: 3.0281 -
accuracy: 0.4185 - val_loss: 11.6184 - val_accuracy: 0.0306
Epoch 60/500
170/170 [==============================] - 5s 32ms/step - loss: 2.9761 -
accuracy: 0.4262 - val_loss: 11.6693 - val_accuracy: 0.0324
Epoch 61/500
170/170 [==============================] - 5s 31ms/step - loss: 2.9316 -
accuracy: 0.4326 - val_loss: 11.7923 - val_accuracy: 0.0315
Epoch 62/500
170/170 [==============================] - 5s 31ms/step - loss: 2.8932 -
accuracy: 0.4399 - val_loss: 11.8787 - val_accuracy: 0.0341
Epoch 63/500
170/170 [==============================] - 5s 31ms/step - loss: 2.8445 -
accuracy: 0.4481 - val_loss: 11.9636 - val_accuracy: 0.0359
Epoch 64/500
170/170 [==============================] - 6s 33ms/step - loss: 2.8043 -
accuracy: 0.4572 - val_loss: 12.0540 - val_accuracy: 0.0332
Epoch 65/500
170/170 [==============================] - 5s 32ms/step - loss: 2.7606 -
accuracy: 0.4607 - val_loss: 12.1173 - val_accuracy: 0.0324
Epoch 66/500
170/170 [==============================] - 5s 32ms/step - loss: 2.7152 -
accuracy: 0.4709 - val_loss: 12.2007 - val_accuracy: 0.0332
```

```
Epoch 67/500
170/170 [==============================] - 5s 31ms/step - loss: 2.6781 -
accuracy: 0.4803 - val_loss: 12.3030 - val_accuracy: 0.0350
Epoch 68/500
170/170 [==============================] - 6s 33ms/step - loss: 2.6325 -
accuracy: 0.4876 - val_loss: 12.3774 - val_accuracy: 0.0332
Epoch 69/500
170/170 [==============================] - 5s 31ms/step - loss: 2.5973 -
accuracy: 0.4936 - val_loss: 12.4287 - val_accuracy: 0.0332
Epoch 70/500
170/170 [==============================] - 6s 33ms/step - loss: 2.5620 -
accuracy: 0.5000 - val_loss: 12.5330 - val_accuracy: 0.0324
Epoch 71/500
170/170 [==============================] - 5s 31ms/step - loss: 2.5206 -
accuracy: 0.5093 - val_loss: 12.5441 - val_accuracy: 0.0350
Epoch 72/500
170/170 [==============================] - 5s 31ms/step - loss: 2.4834 -
accuracy: 0.5186 - val_loss: 12.6802 - val_accuracy: 0.0332
Epoch 73/500
170/170 [==============================] - 5s 32ms/step - loss: 2.4521 -
accuracy: 0.5225 - val_loss: 12.7759 - val_accuracy: 0.0297
Epoch 74/500
170/170 [==============================] - 5s 30ms/step - loss: 2.4129 -
accuracy: 0.5296 - val_loss: 12.8597 - val_accuracy: 0.0350
Epoch 75/500
170/170 [==============================] - 5s 31ms/step - loss: 2.3752 -
accuracy: 0.5351 - val_loss: 12.9296 - val_accuracy: 0.0332
Epoch 76/500
170/170 [==============================] - 5s 32ms/step - loss: 2.3420 -
accuracy: 0.5426 - val_loss: 12.9895 - val_accuracy: 0.0315
Epoch 77/500
170/170 [==============================] - 5s 31ms/step - loss: 2.3070 -
accuracy: 0.5511 - val_loss: 13.0727 - val_accuracy: 0.0306
Epoch 78/500
170/170 [==============================] - 5s 30ms/step - loss: 2.2728 -
accuracy: 0.5590 - val_loss: 13.1265 - val_accuracy: 0.0324
Epoch 79/500
170/170 [==============================] - 5s 32ms/step - loss: 2.2397 -
accuracy: 0.5661 - val_loss: 13.2384 - val_accuracy: 0.0332
Epoch 80/500
170/170 [==============================] - 6s 33ms/step - loss: 2.2063 -
accuracy: 0.5704 - val_loss: 13.2716 - val_accuracy: 0.0341
Epoch 81/500
170/170 [==============================] - 5s 31ms/step - loss: 2.1727 -
accuracy: 0.5771 - val_loss: 13.3678 - val_accuracy: 0.0367
Epoch 82/500
170/170 [==============================] - 5s 31ms/step - loss: 2.1412 -
accuracy: 0.5833 - val_loss: 13.4317 - val_accuracy: 0.0350
```

```
Epoch 83/500
170/170 [==============================] - 5s 30ms/step - loss: 2.1134 -
accuracy: 0.5872 - val_loss: 13.5132 - val_accuracy: 0.0332
Epoch 84/500
170/170 [==============================] - 5s 32ms/step - loss: 2.0823 -
accuracy: 0.5955 - val_loss: 13.5885 - val_accuracy: 0.0324
Epoch 85/500
170/170 [==============================] - 5s 30ms/step - loss: 2.0488 -
accuracy: 0.6021 - val_loss: 13.6575 - val_accuracy: 0.0297
Epoch 86/500
170/170 [==============================] - 5s 32ms/step - loss: 2.0213 -
accuracy: 0.6076 - val_loss: 13.7544 - val_accuracy: 0.0332
Epoch 87/500
170/170 [==============================] - 5s 32ms/step - loss: 1.9935 -
accuracy: 0.6141 - val_loss: 13.7995 - val_accuracy: 0.0350
Epoch 88/500
170/170 [==============================] - 5s 31ms/step - loss: 1.9644 -
accuracy: 0.6176 - val_loss: 13.8952 - val_accuracy: 0.0341
Epoch 89/500
170/170 [==============================] - 5s 32ms/step - loss: 1.9372 -
accuracy: 0.6263 - val_loss: 13.9285 - val_accuracy: 0.0341
Epoch 90/500
170/170 [==============================] - 5s 30ms/step - loss: 1.9111 -
accuracy: 0.6304 - val_loss: 14.0214 - val_accuracy: 0.0332
Epoch 91/500
170/170 [==============================] - 5s 31ms/step - loss: 1.8806 -
accuracy: 0.6380 - val_loss: 14.0616 - val_accuracy: 0.0332
Epoch 92/500
170/170 [==============================] - 5s 31ms/step - loss: 1.8502 -
accuracy: 0.6415 - val_loss: 14.1514 - val_accuracy: 0.0315
Epoch 93/500
170/170 [==============================] - 5s 31ms/step - loss: 1.8291 -
accuracy: 0.6471 - val_loss: 14.2438 - val_accuracy: 0.0341
Epoch 94/500
170/170 [==============================] - 5s 32ms/step - loss: 1.7976 -
accuracy: 0.6535 - val_loss: 14.3174 - val_accuracy: 0.0332
Epoch 95/500
170/170 [==============================] - 5s 30ms/step - loss: 1.7695 -
accuracy: 0.6584 - val_loss: 14.4108 - val_accuracy: 0.0341
Epoch 96/500
170/170 [==============================] - 5s 31ms/step - loss: 1.7419 -
accuracy: 0.6661 - val_loss: 14.4525 - val_accuracy: 0.0341
Epoch 97/500
170/170 [==============================] - 5s 31ms/step - loss: 1.7212 -
accuracy: 0.6701 - val_loss: 14.5836 - val_accuracy: 0.0341
Epoch 98/500
170/170 [==============================] - 5s 30ms/step - loss: 1.6954 -
accuracy: 0.6753 - val_loss: 14.6465 - val_accuracy: 0.0315
```

```
Epoch 99/500
170/170 [==============================] - 5s 31ms/step - loss: 1.6696 -
accuracy: 0.6813 - val_loss: 14.6022 - val_accuracy: 0.0341
Epoch 100/500
170/170 [==============================] - 5s 30ms/step - loss: 1.6429 -
accuracy: 0.6871 - val_loss: 14.7216 - val_accuracy: 0.0376
Epoch 101/500
170/170 [==============================] - 5s 31ms/step - loss: 1.6231 -
accuracy: 0.6908 - val_loss: 14.8235 - val_accuracy: 0.0332
Epoch 102/500
170/170 [==============================] - 5s 32ms/step - loss: 1.5975 -
accuracy: 0.6954 - val_loss: 14.8427 - val_accuracy: 0.0385
Epoch 103/500
170/170 [==============================] - 5s 30ms/step - loss: 1.5745 -
accuracy: 0.7002 - val_loss: 14.8974 - val_accuracy: 0.0350
Epoch 104/500
170/170 [==============================] - 5s 31ms/step - loss: 1.5523 -
accuracy: 0.7060 - val_loss: 14.9321 - val_accuracy: 0.0359
Epoch 105/500
170/170 [==============================] - 5s 31ms/step - loss: 1.5299 -
accuracy: 0.7102 - val_loss: 15.0637 - val_accuracy: 0.0324
Epoch 106/500
170/170 [==============================] - 5s 30ms/step - loss: 1.5079 -
accuracy: 0.7146 - val_loss: 15.0750 - val_accuracy: 0.0324
Epoch 107/500
170/170 [==============================] - 5s 30ms/step - loss: 1.4808 -
accuracy: 0.7201 - val_loss: 15.1970 - val_accuracy: 0.0350
Epoch 108/500
170/170 [==============================] - 5s 30ms/step - loss: 1.4649 -
accuracy: 0.7221 - val_loss: 15.2573 - val_accuracy: 0.0376
Epoch 109/500
170/170 [==============================] - 5s 31ms/step - loss: 1.4422 -
accuracy: 0.7280 - val_loss: 15.3252 - val_accuracy: 0.0341
Epoch 110/500
170/170 [==============================] - 5s 29ms/step - loss: 1.4196 -
accuracy: 0.7338 - val_loss: 15.3678 - val_accuracy: 0.0324
Epoch 111/500
170/170 [==============================] - 5s 31ms/step - loss: 1.4012 -
accuracy: 0.7366 - val_loss: 15.4458 - val_accuracy: 0.0376
Epoch 112/500
170/170 [==============================] - 5s 31ms/step - loss: 1.3784 -
accuracy: 0.7405 - val_loss: 15.4355 - val_accuracy: 0.0341
Epoch 113/500
170/170 [==============================] - 5s 30ms/step - loss: 1.3601 -
accuracy: 0.7451 - val_loss: 15.5646 - val_accuracy: 0.0350
Epoch 114/500
170/170 [==============================] - 5s 30ms/step - loss: 1.3375 -
accuracy: 0.7510 - val_loss: 15.5948 - val_accuracy: 0.0376
```

```
Epoch 115/500
170/170 [==============================] - 5s 31ms/step - loss: 1.3175 -
accuracy: 0.7539 - val_loss: 15.6444 - val_accuracy: 0.0359
Epoch 116/500
170/170 [==============================] - 5s 30ms/step - loss: 1.3004 -
accuracy: 0.7574 - val_loss: 15.7351 - val_accuracy: 0.0324
Epoch 117/500
170/170 [==============================] - 5s 31ms/step - loss: 1.2826 -
accuracy: 0.7626 - val_loss: 15.7707 - val_accuracy: 0.0350
Epoch 118/500
170/170 [==============================] - 5s 30ms/step - loss: 1.2647 -
accuracy: 0.7663 - val_loss: 15.8244 - val_accuracy: 0.0341
Epoch 119/500
170/170 [==============================] - 5s 31ms/step - loss: 1.2493 -
accuracy: 0.7674 - val_loss: 15.9145 - val_accuracy: 0.0359
Epoch 120/500
170/170 [==============================] - 5s 32ms/step - loss: 1.2307 -
accuracy: 0.7721 - val_loss: 15.9446 - val_accuracy: 0.0341
Epoch 121/500
170/170 [==============================] - 5s 29ms/step - loss: 1.2066 -
accuracy: 0.7783 - val_loss: 16.0116 - val_accuracy: 0.0359
Epoch 122/500
170/170 [==============================] - 5s 31ms/step - loss: 1.1892 -
accuracy: 0.7810 - val_loss: 16.0827 - val_accuracy: 0.0332
Epoch 123/500
170/170 [==============================] - 5s 30ms/step - loss: 1.1761 -
accuracy: 0.7839 - val_loss: 16.0970 - val_accuracy: 0.0332
Epoch 124/500
170/170 [==============================] - 5s 31ms/step - loss: 1.1546 -
accuracy: 0.7885 - val_loss: 16.2138 - val_accuracy: 0.0332
Epoch 125/500
170/170 [==============================] - 5s 30ms/step - loss: 1.1453 -
accuracy: 0.7918 - val_loss: 16.2405 - val_accuracy: 0.0341
Epoch 126/500
170/170 [==============================] - 5s 31ms/step - loss: 1.1288 -
accuracy: 0.7937 - val_loss: 16.2552 - val_accuracy: 0.0350
Epoch 127/500
170/170 [==============================] - 5s 30ms/step - loss: 1.1037 -
accuracy: 0.7998 - val_loss: 16.3812 - val_accuracy: 0.0324
Epoch 128/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0893 -
accuracy: 0.8015 - val_loss: 16.3964 - val_accuracy: 0.0315
Epoch 129/500
170/170 [==============================] - 5s 30ms/step - loss: 1.0733 -
accuracy: 0.8072 - val_loss: 16.4773 - val_accuracy: 0.0367
Epoch 130/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0646 -
accuracy: 0.8057 - val_loss: 16.5490 - val_accuracy: 0.0315
```

```
Epoch 131/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0455 -
accuracy: 0.8109 - val_loss: 16.5184 - val_accuracy: 0.0332
Epoch 132/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0303 -
accuracy: 0.8164 - val_loss: 16.5754 - val_accuracy: 0.0332
Epoch 133/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0140 -
accuracy: 0.8201 - val_loss: 16.6357 - val_accuracy: 0.0324
Epoch 134/500
170/170 [==============================] - 5s 31ms/step - loss: 1.0012 -
accuracy: 0.8211 - val_loss: 16.6827 - val_accuracy: 0.0376
Epoch 135/500
170/170 [==============================] - 5s 30ms/step - loss: 0.9896 -
accuracy: 0.8246 - val_loss: 16.7537 - val_accuracy: 0.0359
Epoch 136/500
170/170 [==============================] - 5s 31ms/step - loss: 0.9727 -
accuracy: 0.8260 - val_loss: 16.7713 - val_accuracy: 0.0324
Epoch 137/500
170/170 [==============================] - 5s 30ms/step - loss: 0.9593 -
accuracy: 0.8287 - val_loss: 16.8508 - val_accuracy: 0.0385
Epoch 138/500
170/170 [==============================] - 5s 31ms/step - loss: 0.9479 -
accuracy: 0.8306 - val_loss: 16.9070 - val_accuracy: 0.0332
Epoch 139/500
170/170 [==============================] - 5s 30ms/step - loss: 0.9351 -
accuracy: 0.8329 - val_loss: 16.9600 - val_accuracy: 0.0332
Epoch 140/500
170/170 [==============================] - 5s 29ms/step - loss: 0.9165 -
accuracy: 0.8363 - val_loss: 16.9916 - val_accuracy: 0.0324
Epoch 141/500
170/170 [==============================] - 5s 30ms/step - loss: 0.9038 -
accuracy: 0.8404 - val_loss: 17.0682 - val_accuracy: 0.0367
Epoch 142/500
170/170 [==============================] - 5s 30ms/step - loss: 0.8932 -
accuracy: 0.8424 - val_loss: 17.1283 - val_accuracy: 0.0332
Epoch 143/500
170/170 [==============================] - 5s 30ms/step - loss: 0.8830 -
accuracy: 0.8435 - val_loss: 17.1977 - val_accuracy: 0.0315
Epoch 144/500
170/170 [==============================] - 5s 31ms/step - loss: 0.8676 -
accuracy: 0.8471 - val_loss: 17.1735 - val_accuracy: 0.0367
Epoch 145/500
170/170 [==============================] - 5s 29ms/step - loss: 0.8512 -
accuracy: 0.8497 - val_loss: 17.2491 - val_accuracy: 0.0324
Epoch 146/500
170/170 [==============================] - 5s 29ms/step - loss: 0.8437 -
accuracy: 0.8503 - val_loss: 17.2875 - val_accuracy: 0.0332
```

```
Epoch 147/500
170/170 [==============================] - 5s 31ms/step - loss: 0.8321 -
accuracy: 0.8539 - val_loss: 17.3817 - val_accuracy: 0.0324
Epoch 148/500
170/170 [==============================] - 5s 30ms/step - loss: 0.8186 -
accuracy: 0.8566 - val_loss: 17.4105 - val_accuracy: 0.0367
Epoch 149/500
170/170 [==============================] - 5s 29ms/step - loss: 0.8048 -
accuracy: 0.8596 - val_loss: 17.4207 - val_accuracy: 0.0315
Epoch 150/500
170/170 [==============================] - 5s 31ms/step - loss: 0.7937 -
accuracy: 0.8606 - val_loss: 17.4985 - val_accuracy: 0.0350
Epoch 151/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7836 -
accuracy: 0.8638 - val_loss: 17.5255 - val_accuracy: 0.0280
Epoch 152/500
170/170 [==============================] - 5s 31ms/step - loss: 0.7805 -
accuracy: 0.8647 - val_loss: 17.5783 - val_accuracy: 0.0315
Epoch 153/500
170/170 [==============================] - 5s 31ms/step - loss: 0.7665 -
accuracy: 0.8651 - val_loss: 17.5689 - val_accuracy: 0.0332
Epoch 154/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7553 -
accuracy: 0.8675 - val_loss: 17.6418 - val_accuracy: 0.0306
Epoch 155/500
170/170 [==============================] - 5s 31ms/step - loss: 0.7420 -
accuracy: 0.8708 - val_loss: 17.7427 - val_accuracy: 0.0341
Epoch 156/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7285 -
accuracy: 0.8733 - val_loss: 17.7536 - val_accuracy: 0.0341
Epoch 157/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7145 -
accuracy: 0.8754 - val_loss: 17.8108 - val_accuracy: 0.0280
Epoch 158/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7100 -
accuracy: 0.8753 - val_loss: 17.8222 - val_accuracy: 0.0341
Epoch 159/500
170/170 [==============================] - 5s 30ms/step - loss: 0.7102 -
accuracy: 0.8770 - val_loss: 17.8850 - val_accuracy: 0.0332
Epoch 160/500
170/170 [==============================] - 5s 31ms/step - loss: 0.7040 -
accuracy: 0.8769 - val_loss: 17.8911 - val_accuracy: 0.0350
Epoch 161/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6948 -
accuracy: 0.8784 - val_loss: 18.0001 - val_accuracy: 0.0341
Epoch 162/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6886 -
accuracy: 0.8776 - val_loss: 17.9406 - val_accuracy: 0.0315
```

```
Epoch 163/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6685 -
accuracy: 0.8835 - val_loss: 18.0804 - val_accuracy: 0.0350
Epoch 164/500
170/170 [==============================] - 5s 30ms/step - loss: 0.6498 -
accuracy: 0.8870 - val_loss: 18.0691 - val_accuracy: 0.0315
Epoch 165/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6378 -
accuracy: 0.8885 - val_loss: 18.0945 - val_accuracy: 0.0297
Epoch 166/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6340 -
accuracy: 0.8889 - val_loss: 18.1534 - val_accuracy: 0.0297
Epoch 167/500
170/170 [==============================] - 5s 31ms/step - loss: 0.6223 -
accuracy: 0.8911 - val_loss: 18.1886 - val_accuracy: 0.0332
Epoch 168/500
170/170 [==============================] - 5s 30ms/step - loss: 0.6175 -
accuracy: 0.8930 - val_loss: 18.2433 - val_accuracy: 0.0306
Epoch 169/500
170/170 [==============================] - 5s 32ms/step - loss: 0.6156 -
accuracy: 0.8927 - val_loss: 18.3182 - val_accuracy: 0.0315
Epoch 170/500
170/170 [==============================] - 5s 30ms/step - loss: 0.6046 -
accuracy: 0.8946 - val_loss: 18.2726 - val_accuracy: 0.0297
Epoch 171/500
170/170 [==============================] - 5s 30ms/step - loss: 0.6026 -
accuracy: 0.8952 - val_loss: 18.4143 - val_accuracy: 0.0306
Epoch 172/500
170/170 [==============================] - 5s 32ms/step - loss: 0.5933 -
accuracy: 0.8968 - val_loss: 18.4218 - val_accuracy: 0.0306
Epoch 173/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5935 -
accuracy: 0.8955 - val_loss: 18.4620 - val_accuracy: 0.0332
Epoch 174/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5845 -
accuracy: 0.8971 - val_loss: 18.4606 - val_accuracy: 0.0324
Epoch 175/500
170/170 [==============================] - 5s 31ms/step - loss: 0.5779 -
accuracy: 0.8986 - val_loss: 18.4733 - val_accuracy: 0.0289
Epoch 176/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5635 -
accuracy: 0.9005 - val_loss: 18.5593 - val_accuracy: 0.0289
Epoch 177/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5519 -
accuracy: 0.9035 - val_loss: 18.5730 - val_accuracy: 0.0289
Epoch 178/500
170/170 [==============================] - 5s 31ms/step - loss: 0.5528 -
accuracy: 0.9027 - val_loss: 18.6487 - val_accuracy: 0.0324
```

```
Epoch 179/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5441 -
accuracy: 0.9044 - val_loss: 18.6678 - val_accuracy: 0.0324
Epoch 180/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5330 -
accuracy: 0.9064 - val_loss: 18.7264 - val_accuracy: 0.0324
Epoch 181/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5275 -
accuracy: 0.9068 - val_loss: 18.7117 - val_accuracy: 0.0297
Epoch 182/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5200 -
accuracy: 0.9085 - val_loss: 18.8183 - val_accuracy: 0.0306
Epoch 183/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5127 -
accuracy: 0.9092 - val_loss: 18.7757 - val_accuracy: 0.0306
Epoch 184/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5056 -
accuracy: 0.9115 - val_loss: 18.8401 - val_accuracy: 0.0324
Epoch 185/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4972 -
accuracy: 0.9126 - val_loss: 18.9193 - val_accuracy: 0.0306
Epoch 186/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4909 -
accuracy: 0.9143 - val_loss: 18.8788 - val_accuracy: 0.0315
Epoch 187/500
170/170 [==============================] - 5s 31ms/step - loss: 0.5002 -
accuracy: 0.9122 - val_loss: 18.9402 - val_accuracy: 0.0289
Epoch 188/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5020 -
accuracy: 0.9108 - val_loss: 18.9864 - val_accuracy: 0.0306
Epoch 189/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4971 -
accuracy: 0.9125 - val_loss: 19.0146 - val_accuracy: 0.0315
Epoch 190/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4923 -
accuracy: 0.9122 - val_loss: 19.0894 - val_accuracy: 0.0315
Epoch 191/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4749 -
accuracy: 0.9150 - val_loss: 19.0643 - val_accuracy: 0.0306
Epoch 192/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4650 -
accuracy: 0.9178 - val_loss: 19.1478 - val_accuracy: 0.0289
Epoch 193/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4535 -
accuracy: 0.9178 - val_loss: 19.1879 - val_accuracy: 0.0289
Epoch 194/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4488 -
accuracy: 0.9201 - val_loss: 19.1793 - val_accuracy: 0.0297
```

```
Epoch 195/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4383 -
accuracy: 0.9200 - val_loss: 19.2190 - val_accuracy: 0.0289
Epoch 196/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4399 -
accuracy: 0.9201 - val_loss: 19.2529 - val_accuracy: 0.0306
Epoch 197/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4464 -
accuracy: 0.9191 - val_loss: 19.3211 - val_accuracy: 0.0315
Epoch 198/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4550 -
accuracy: 0.9166 - val_loss: 19.3222 - val_accuracy: 0.0306
Epoch 199/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4479 -
accuracy: 0.9196 - val_loss: 19.3504 - val_accuracy: 0.0306
Epoch 200/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4474 -
accuracy: 0.9184 - val_loss: 19.4139 - val_accuracy: 0.0289
Epoch 201/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4269 -
accuracy: 0.9220 - val_loss: 19.4351 - val_accuracy: 0.0315
Epoch 202/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4109 -
accuracy: 0.9245 - val_loss: 19.5064 - val_accuracy: 0.0262
Epoch 203/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4058 -
accuracy: 0.9242 - val_loss: 19.5058 - val_accuracy: 0.0306
Epoch 204/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4027 -
accuracy: 0.9247 - val_loss: 19.5502 - val_accuracy: 0.0306
Epoch 205/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3935 -
accuracy: 0.9265 - val_loss: 19.5602 - val_accuracy: 0.0315
Epoch 206/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3899 -
accuracy: 0.9261 - val_loss: 19.5809 - val_accuracy: 0.0332
Epoch 207/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4037 -
accuracy: 0.9241 - val_loss: 19.5657 - val_accuracy: 0.0341
Epoch 208/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4239 -
accuracy: 0.9205 - val_loss: 19.6497 - val_accuracy: 0.0271
Epoch 209/500
170/170 [==============================] - 5s 32ms/step - loss: 0.4403 -
accuracy: 0.9176 - val_loss: 19.6919 - val_accuracy: 0.0306
Epoch 210/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4443 -
accuracy: 0.9166 - val_loss: 19.7702 - val_accuracy: 0.0306
```

```
Epoch 211/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4164 -
accuracy: 0.9225 - val_loss: 19.7960 - val_accuracy: 0.0280
Epoch 212/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3861 -
accuracy: 0.9270 - val_loss: 19.8382 - val_accuracy: 0.0297
Epoch 213/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3650 -
accuracy: 0.9303 - val_loss: 19.9013 - val_accuracy: 0.0306
Epoch 214/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3973 -
accuracy: 0.9228 - val_loss: 19.9649 - val_accuracy: 0.0280
Epoch 215/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3680 -
accuracy: 0.9288 - val_loss: 20.0002 - val_accuracy: 0.0315
Epoch 216/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3524 -
accuracy: 0.9307 - val_loss: 20.0537 - val_accuracy: 0.0332
Epoch 217/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3489 -
accuracy: 0.9314 - val_loss: 20.0807 - val_accuracy: 0.0297
Epoch 218/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3448 -
accuracy: 0.9317 - val_loss: 20.0816 - val_accuracy: 0.0306
Epoch 219/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3437 -
accuracy: 0.9313 - val_loss: 20.0964 - val_accuracy: 0.0315
Epoch 220/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3417 -
accuracy: 0.9318 - val_loss: 20.1243 - val_accuracy: 0.0332
Epoch 221/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3383 -
accuracy: 0.9318 - val_loss: 20.1806 - val_accuracy: 0.0280
Epoch 222/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3446 -
accuracy: 0.9316 - val_loss: 20.2099 - val_accuracy: 0.0297
Epoch 223/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3835 -
accuracy: 0.9244 - val_loss: 20.1490 - val_accuracy: 0.0315
Epoch 224/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4821 -
accuracy: 0.9018 - val_loss: 20.1611 - val_accuracy: 0.0289
Epoch 225/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4695 -
accuracy: 0.9061 - val_loss: 20.1732 - val_accuracy: 0.0280
Epoch 226/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3734 -
accuracy: 0.9275 - val_loss: 20.2320 - val_accuracy: 0.0280
```

```
Epoch 227/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3355 -
accuracy: 0.9330 - val_loss: 20.4001 - val_accuracy: 0.0262
Epoch 228/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3221 -
accuracy: 0.9347 - val_loss: 20.3895 - val_accuracy: 0.0271
Epoch 229/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3158 -
accuracy: 0.9354 - val_loss: 20.4033 - val_accuracy: 0.0271
Epoch 230/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3131 -
accuracy: 0.9342 - val_loss: 20.4424 - val_accuracy: 0.0289
Epoch 231/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3111 -
accuracy: 0.9348 - val_loss: 20.4563 - val_accuracy: 0.0271
Epoch 232/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3080 -
accuracy: 0.9356 - val_loss: 20.4634 - val_accuracy: 0.0280
Epoch 233/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3064 -
accuracy: 0.9360 - val_loss: 20.5328 - val_accuracy: 0.0289
Epoch 234/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3053 -
accuracy: 0.9350 - val_loss: 20.5510 - val_accuracy: 0.0289
Epoch 235/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3046 -
accuracy: 0.9359 - val_loss: 20.5697 - val_accuracy: 0.0289
Epoch 236/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3040 -
accuracy: 0.9354 - val_loss: 20.6020 - val_accuracy: 0.0289
Epoch 237/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3038 -
accuracy: 0.9354 - val_loss: 20.5621 - val_accuracy: 0.0280
Epoch 238/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3032 -
accuracy: 0.9353 - val_loss: 20.6278 - val_accuracy: 0.0315
Epoch 239/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3035 -
accuracy: 0.9353 - val_loss: 20.6332 - val_accuracy: 0.0280
Epoch 240/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3657 -
accuracy: 0.9233 - val_loss: 20.5032 - val_accuracy: 0.0280
Epoch 241/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4879 -
accuracy: 0.8942 - val_loss: 20.6235 - val_accuracy: 0.0297
Epoch 242/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4119 -
accuracy: 0.9153 - val_loss: 20.6551 - val_accuracy: 0.0289
```

```
Epoch 243/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3242 -
accuracy: 0.9331 - val_loss: 20.7189 - val_accuracy: 0.0306
Epoch 244/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2966 -
accuracy: 0.9366 - val_loss: 20.7340 - val_accuracy: 0.0289
Epoch 245/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2883 -
accuracy: 0.9375 - val_loss: 20.7586 - val_accuracy: 0.0289
Epoch 246/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2845 -
accuracy: 0.9371 - val_loss: 20.7597 - val_accuracy: 0.0289
Epoch 247/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2820 -
accuracy: 0.9366 - val_loss: 20.8042 - val_accuracy: 0.0262
Epoch 248/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2799 -
accuracy: 0.9373 - val_loss: 20.8204 - val_accuracy: 0.0245
Epoch 249/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2786 -
accuracy: 0.9374 - val_loss: 20.8163 - val_accuracy: 0.0289
Epoch 250/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2772 -
accuracy: 0.9366 - val_loss: 20.8278 - val_accuracy: 0.0297
Epoch 251/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2761 -
accuracy: 0.9375 - val_loss: 20.8472 - val_accuracy: 0.0262
Epoch 252/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2757 -
accuracy: 0.9371 - val_loss: 20.8607 - val_accuracy: 0.0262
Epoch 253/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2743 -
accuracy: 0.9382 - val_loss: 20.9086 - val_accuracy: 0.0297
Epoch 254/500
170/170 [==============================] - 5s 32ms/step - loss: 0.2726 -
accuracy: 0.9375 - val_loss: 20.9217 - val_accuracy: 0.0271
Epoch 255/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2718 -
accuracy: 0.9375 - val_loss: 20.9152 - val_accuracy: 0.0306
Epoch 256/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2730 -
accuracy: 0.9377 - val_loss: 20.9142 - val_accuracy: 0.0315
Epoch 257/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2835 -
accuracy: 0.9361 - val_loss: 20.9337 - val_accuracy: 0.0271
Epoch 258/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3726 -
accuracy: 0.9188 - val_loss: 20.7737 - val_accuracy: 0.0289
```

```
Epoch 259/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4769 -
accuracy: 0.8904 - val_loss: 21.0212 - val_accuracy: 0.0297
Epoch 260/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3535 -
accuracy: 0.9248 - val_loss: 20.9244 - val_accuracy: 0.0306
Epoch 261/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2884 -
accuracy: 0.9366 - val_loss: 21.0479 - val_accuracy: 0.0306
Epoch 262/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2691 -
accuracy: 0.9386 - val_loss: 21.0450 - val_accuracy: 0.0280
Epoch 263/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2636 -
accuracy: 0.9382 - val_loss: 21.0375 - val_accuracy: 0.0289
Epoch 264/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2611 -
accuracy: 0.9389 - val_loss: 21.0916 - val_accuracy: 0.0289
Epoch 265/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2596 -
accuracy: 0.9382 - val_loss: 21.0988 - val_accuracy: 0.0289
Epoch 266/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2576 -
accuracy: 0.9385 - val_loss: 21.0983 - val_accuracy: 0.0271
Epoch 267/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2568 -
accuracy: 0.9388 - val_loss: 21.1275 - val_accuracy: 0.0262
Epoch 268/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2559 -
accuracy: 0.9394 - val_loss: 21.1828 - val_accuracy: 0.0306
Epoch 269/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2544 -
accuracy: 0.9388 - val_loss: 21.1459 - val_accuracy: 0.0262
Epoch 270/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2558 -
accuracy: 0.9387 - val_loss: 21.2061 - val_accuracy: 0.0289
Epoch 271/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2537 -
accuracy: 0.9396 - val_loss: 21.2214 - val_accuracy: 0.0306
Epoch 272/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2533 -
accuracy: 0.9378 - val_loss: 21.2612 - val_accuracy: 0.0297
Epoch 273/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2526 -
accuracy: 0.9388 - val_loss: 21.2573 - val_accuracy: 0.0306
Epoch 274/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2527 -
accuracy: 0.9386 - val_loss: 21.2847 - val_accuracy: 0.0306
```

```
Epoch 275/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2529 -
accuracy: 0.9383 - val_loss: 21.3322 - val_accuracy: 0.0271
Epoch 276/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2725 -
accuracy: 0.9349 - val_loss: 21.4449 - val_accuracy: 0.0280
Epoch 277/500
170/170 [==============================] - 5s 30ms/step - loss: 0.5157 -
accuracy: 0.8767 - val_loss: 21.2014 - val_accuracy: 0.0236
Epoch 278/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4249 -
accuracy: 0.9022 - val_loss: 21.2554 - val_accuracy: 0.0306
Epoch 279/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2971 -
accuracy: 0.9329 - val_loss: 21.3807 - val_accuracy: 0.0271
Epoch 280/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2591 -
accuracy: 0.9386 - val_loss: 21.4715 - val_accuracy: 0.0297
Epoch 281/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2497 -
accuracy: 0.9394 - val_loss: 21.4866 - val_accuracy: 0.0315
Epoch 282/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2461 -
accuracy: 0.9398 - val_loss: 21.5283 - val_accuracy: 0.0289
Epoch 283/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2443 -
accuracy: 0.9400 - val_loss: 21.5117 - val_accuracy: 0.0306
Epoch 284/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2431 -
accuracy: 0.9392 - val_loss: 21.5611 - val_accuracy: 0.0297
Epoch 285/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2420 -
accuracy: 0.9396 - val_loss: 21.5650 - val_accuracy: 0.0289
Epoch 286/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2416 -
accuracy: 0.9390 - val_loss: 21.5685 - val_accuracy: 0.0306
Epoch 287/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2408 -
accuracy: 0.9389 - val_loss: 21.5728 - val_accuracy: 0.0306
Epoch 288/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2400 -
accuracy: 0.9389 - val_loss: 21.6174 - val_accuracy: 0.0306
Epoch 289/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2388 -
accuracy: 0.9390 - val_loss: 21.6229 - val_accuracy: 0.0306
Epoch 290/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2381 -
accuracy: 0.9392 - val_loss: 21.6159 - val_accuracy: 0.0280
```

```
Epoch 291/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2376 -
accuracy: 0.9390 - val_loss: 21.6681 - val_accuracy: 0.0315
Epoch 292/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2386 -
accuracy: 0.9390 - val_loss: 21.7086 - val_accuracy: 0.0306
Epoch 293/500
170/170 [==============================] - 6s 33ms/step - loss: 0.2392 -
accuracy: 0.9391 - val_loss: 21.7033 - val_accuracy: 0.0324
Epoch 294/500
170/170 [==============================] - 5s 32ms/step - loss: 0.2380 -
accuracy: 0.9388 - val_loss: 21.7336 - val_accuracy: 0.0315
Epoch 295/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2378 -
accuracy: 0.9388 - val_loss: 21.7202 - val_accuracy: 0.0315
Epoch 296/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2501 -
accuracy: 0.9371 - val_loss: 21.6938 - val_accuracy: 0.0271
Epoch 297/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4727 -
accuracy: 0.8832 - val_loss: 21.6152 - val_accuracy: 0.0262
Epoch 298/500
170/170 [==============================] - 5s 32ms/step - loss: 0.4078 -
accuracy: 0.9026 - val_loss: 21.6270 - val_accuracy: 0.0297
Epoch 299/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2839 -
accuracy: 0.9336 - val_loss: 21.6250 - val_accuracy: 0.0289
Epoch 300/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2469 -
accuracy: 0.9393 - val_loss: 21.7058 - val_accuracy: 0.0315
Epoch 301/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2367 -
accuracy: 0.9398 - val_loss: 21.7012 - val_accuracy: 0.0306
Epoch 302/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2340 -
accuracy: 0.9395 - val_loss: 21.7367 - val_accuracy: 0.0297
Epoch 303/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2323 -
accuracy: 0.9398 - val_loss: 21.7778 - val_accuracy: 0.0289
Epoch 304/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2314 -
accuracy: 0.9394 - val_loss: 21.7691 - val_accuracy: 0.0324
Epoch 305/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2307 -
accuracy: 0.9398 - val_loss: 21.7999 - val_accuracy: 0.0297
Epoch 306/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2299 -
accuracy: 0.9388 - val_loss: 21.7645 - val_accuracy: 0.0289
```

```
Epoch 307/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2291 -
accuracy: 0.9389 - val_loss: 21.7837 - val_accuracy: 0.0297
Epoch 308/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2285 -
accuracy: 0.9389 - val_loss: 21.8158 - val_accuracy: 0.0306
Epoch 309/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2275 -
accuracy: 0.9390 - val_loss: 21.8224 - val_accuracy: 0.0306
Epoch 310/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2279 -
accuracy: 0.9387 - val_loss: 21.8785 - val_accuracy: 0.0297
Epoch 311/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2274 -
accuracy: 0.9394 - val_loss: 21.8765 - val_accuracy: 0.0289
Epoch 312/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2270 -
accuracy: 0.9394 - val_loss: 21.8697 - val_accuracy: 0.0297
Epoch 313/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2274 -
accuracy: 0.9393 - val_loss: 21.8733 - val_accuracy: 0.0289
Epoch 314/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2261 -
accuracy: 0.9401 - val_loss: 21.9034 - val_accuracy: 0.0297
Epoch 315/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2266 -
accuracy: 0.9401 - val_loss: 21.9591 - val_accuracy: 0.0306
Epoch 316/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2674 -
accuracy: 0.9325 - val_loss: 21.8130 - val_accuracy: 0.0271
Epoch 317/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4603 -
accuracy: 0.8831 - val_loss: 21.8556 - val_accuracy: 0.0289
Epoch 318/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3422 -
accuracy: 0.9168 - val_loss: 21.9408 - val_accuracy: 0.0271
Epoch 319/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2609 -
accuracy: 0.9360 - val_loss: 21.9806 - val_accuracy: 0.0289
Epoch 320/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2328 -
accuracy: 0.9397 - val_loss: 22.0415 - val_accuracy: 0.0306
Epoch 321/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2263 -
accuracy: 0.9401 - val_loss: 22.0575 - val_accuracy: 0.0297
Epoch 322/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2245 -
accuracy: 0.9398 - val_loss: 22.0642 - val_accuracy: 0.0297
```

```
Epoch 323/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2230 -
accuracy: 0.9392 - val_loss: 22.1055 - val_accuracy: 0.0306
Epoch 324/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2217 -
accuracy: 0.9402 - val_loss: 22.0849 - val_accuracy: 0.0271
Epoch 325/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2215 -
accuracy: 0.9398 - val_loss: 22.0922 - val_accuracy: 0.0297
Epoch 326/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2212 -
accuracy: 0.9394 - val_loss: 22.0894 - val_accuracy: 0.0280
Epoch 327/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2206 -
accuracy: 0.9392 - val_loss: 22.1556 - val_accuracy: 0.0297
Epoch 328/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2201 -
accuracy: 0.9405 - val_loss: 22.1446 - val_accuracy: 0.0306
Epoch 329/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2199 -
accuracy: 0.9390 - val_loss: 22.1392 - val_accuracy: 0.0289
Epoch 330/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2192 -
accuracy: 0.9394 - val_loss: 22.1586 - val_accuracy: 0.0297
Epoch 331/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2189 -
accuracy: 0.9390 - val_loss: 22.1744 - val_accuracy: 0.0315
Epoch 332/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2186 -
accuracy: 0.9399 - val_loss: 22.1339 - val_accuracy: 0.0306
Epoch 333/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2187 -
accuracy: 0.9393 - val_loss: 22.1781 - val_accuracy: 0.0297
Epoch 334/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2183 -
accuracy: 0.9394 - val_loss: 22.2224 - val_accuracy: 0.0289
Epoch 335/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2186 -
accuracy: 0.9399 - val_loss: 22.2066 - val_accuracy: 0.0289
Epoch 336/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2192 -
accuracy: 0.9400 - val_loss: 22.2165 - val_accuracy: 0.0297
Epoch 337/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2209 -
accuracy: 0.9386 - val_loss: 22.2347 - val_accuracy: 0.0289
Epoch 338/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3193 -
accuracy: 0.9162 - val_loss: 22.1021 - val_accuracy: 0.0341
```

```
Epoch 339/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4737 -
accuracy: 0.8782 - val_loss: 22.0650 - val_accuracy: 0.0315
Epoch 340/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2991 -
accuracy: 0.9243 - val_loss: 22.1199 - val_accuracy: 0.0324
Epoch 341/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2367 -
accuracy: 0.9386 - val_loss: 22.2150 - val_accuracy: 0.0324
Epoch 342/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2207 -
accuracy: 0.9401 - val_loss: 22.2393 - val_accuracy: 0.0315
Epoch 343/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2178 -
accuracy: 0.9401 - val_loss: 22.2745 - val_accuracy: 0.0315
Epoch 344/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2160 -
accuracy: 0.9398 - val_loss: 22.2823 - val_accuracy: 0.0324
Epoch 345/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2152 -
accuracy: 0.9409 - val_loss: 22.2811 - val_accuracy: 0.0306
Epoch 346/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2154 -
accuracy: 0.9397 - val_loss: 22.2924 - val_accuracy: 0.0297
Epoch 347/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2143 -
accuracy: 0.9397 - val_loss: 22.3388 - val_accuracy: 0.0306
Epoch 348/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2134 -
accuracy: 0.9400 - val_loss: 22.3096 - val_accuracy: 0.0306
Epoch 349/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2131 -
accuracy: 0.9396 - val_loss: 22.3222 - val_accuracy: 0.0289
Epoch 350/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2133 -
accuracy: 0.9395 - val_loss: 22.3506 - val_accuracy: 0.0306
Epoch 351/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2127 -
accuracy: 0.9399 - val_loss: 22.3535 - val_accuracy: 0.0297
Epoch 352/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2128 -
accuracy: 0.9395 - val_loss: 22.3715 - val_accuracy: 0.0297
Epoch 353/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2125 -
accuracy: 0.9397 - val_loss: 22.4155 - val_accuracy: 0.0306
Epoch 354/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2121 -
accuracy: 0.9390 - val_loss: 22.4637 - val_accuracy: 0.0306
```

```
Epoch 355/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2122 -
accuracy: 0.9392 - val_loss: 22.4329 - val_accuracy: 0.0324
Epoch 356/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2124 -
accuracy: 0.9394 - val_loss: 22.4721 - val_accuracy: 0.0306
Epoch 357/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2133 -
accuracy: 0.9399 - val_loss: 22.5121 - val_accuracy: 0.0297
Epoch 358/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2129 -
accuracy: 0.9396 - val_loss: 22.4971 - val_accuracy: 0.0289
Epoch 359/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2269 -
accuracy: 0.9379 - val_loss: 22.5104 - val_accuracy: 0.0315
Epoch 360/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3655 -
accuracy: 0.9051 - val_loss: 22.5814 - val_accuracy: 0.0297
Epoch 361/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3825 -
accuracy: 0.8997 - val_loss: 22.2862 - val_accuracy: 0.0280
Epoch 362/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2619 -
accuracy: 0.9327 - val_loss: 22.4455 - val_accuracy: 0.0289
Epoch 363/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2227 -
accuracy: 0.9393 - val_loss: 22.4724 - val_accuracy: 0.0297
Epoch 364/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2138 -
accuracy: 0.9398 - val_loss: 22.4566 - val_accuracy: 0.0289
Epoch 365/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2121 -
accuracy: 0.9400 - val_loss: 22.4846 - val_accuracy: 0.0280
Epoch 366/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2103 -
accuracy: 0.9397 - val_loss: 22.4838 - val_accuracy: 0.0297
Epoch 367/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2096 -
accuracy: 0.9410 - val_loss: 22.4877 - val_accuracy: 0.0297
Epoch 368/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2091 -
accuracy: 0.9399 - val_loss: 22.5093 - val_accuracy: 0.0306
Epoch 369/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2085 -
accuracy: 0.9392 - val_loss: 22.4916 - val_accuracy: 0.0297
Epoch 370/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2086 -
accuracy: 0.9394 - val_loss: 22.5368 - val_accuracy: 0.0297
```

```
Epoch 371/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2082 -
accuracy: 0.9397 - val_loss: 22.5126 - val_accuracy: 0.0297
Epoch 372/500
170/170 [==============================] - 5s 32ms/step - loss: 0.2075 -
accuracy: 0.9399 - val_loss: 22.5533 - val_accuracy: 0.0306
Epoch 373/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2077 -
accuracy: 0.9398 - val_loss: 22.5657 - val_accuracy: 0.0297
Epoch 374/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2076 -
accuracy: 0.9400 - val_loss: 22.5953 - val_accuracy: 0.0297
Epoch 375/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2073 -
accuracy: 0.9396 - val_loss: 22.5938 - val_accuracy: 0.0289
Epoch 376/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2071 -
accuracy: 0.9397 - val_loss: 22.6470 - val_accuracy: 0.0271
Epoch 377/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2072 -
accuracy: 0.9396 - val_loss: 22.6282 - val_accuracy: 0.0289
Epoch 378/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2074 -
accuracy: 0.9391 - val_loss: 22.6293 - val_accuracy: 0.0306
Epoch 379/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2086 -
accuracy: 0.9400 - val_loss: 22.6557 - val_accuracy: 0.0297
Epoch 380/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2133 -
accuracy: 0.9385 - val_loss: 22.6633 - val_accuracy: 0.0297
Epoch 381/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2464 -
accuracy: 0.9329 - val_loss: 22.7099 - val_accuracy: 0.0297
Epoch 382/500
170/170 [==============================] - 5s 30ms/step - loss: 0.4143 -
accuracy: 0.8916 - val_loss: 22.5461 - val_accuracy: 0.0297
Epoch 383/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3039 -
accuracy: 0.9201 - val_loss: 22.6728 - val_accuracy: 0.0315
Epoch 384/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2320 -
accuracy: 0.9370 - val_loss: 22.6848 - val_accuracy: 0.0315
Epoch 385/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2125 -
accuracy: 0.9398 - val_loss: 22.6765 - val_accuracy: 0.0324
Epoch 386/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2081 -
accuracy: 0.9399 - val_loss: 22.6996 - val_accuracy: 0.0324
```

```
Epoch 387/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2072 -
accuracy: 0.9402 - val_loss: 22.7072 - val_accuracy: 0.0315
Epoch 388/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2056 -
accuracy: 0.9400 - val_loss: 22.7123 - val_accuracy: 0.0332
Epoch 389/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2055 -
accuracy: 0.9398 - val_loss: 22.6939 - val_accuracy: 0.0324
Epoch 390/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2049 -
accuracy: 0.9399 - val_loss: 22.7147 - val_accuracy: 0.0324
Epoch 391/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2044 -
accuracy: 0.9401 - val_loss: 22.7316 - val_accuracy: 0.0341
Epoch 392/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2048 -
accuracy: 0.9396 - val_loss: 22.7177 - val_accuracy: 0.0332
Epoch 393/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2047 -
accuracy: 0.9390 - val_loss: 22.7651 - val_accuracy: 0.0324
Epoch 394/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2036 -
accuracy: 0.9393 - val_loss: 22.7330 - val_accuracy: 0.0306
Epoch 395/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2040 -
accuracy: 0.9398 - val_loss: 22.7304 - val_accuracy: 0.0332
Epoch 396/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2039 -
accuracy: 0.9392 - val_loss: 22.7569 - val_accuracy: 0.0324
Epoch 397/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2034 -
accuracy: 0.9401 - val_loss: 22.7523 - val_accuracy: 0.0289
Epoch 398/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2037 -
accuracy: 0.9393 - val_loss: 22.7872 - val_accuracy: 0.0315
Epoch 399/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2034 -
accuracy: 0.9395 - val_loss: 22.8704 - val_accuracy: 0.0324
Epoch 400/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2035 -
accuracy: 0.9397 - val_loss: 22.8402 - val_accuracy: 0.0306
Epoch 401/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2035 -
accuracy: 0.9397 - val_loss: 22.8452 - val_accuracy: 0.0315
Epoch 402/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2038 -
accuracy: 0.9390 - val_loss: 22.8474 - val_accuracy: 0.0297
```

```
Epoch 403/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2042 -
accuracy: 0.9401 - val_loss: 22.8572 - val_accuracy: 0.0315
Epoch 404/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2113 -
accuracy: 0.9386 - val_loss: 22.8889 - val_accuracy: 0.0306
Epoch 405/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3200 -
accuracy: 0.9130 - val_loss: 22.8660 - val_accuracy: 0.0306
Epoch 406/500
170/170 [==============================] - 5s 31ms/step - loss: 0.4457 -
accuracy: 0.8807 - val_loss: 22.8219 - val_accuracy: 0.0315
Epoch 407/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2669 -
accuracy: 0.9290 - val_loss: 22.9144 - val_accuracy: 0.0315
Epoch 408/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2173 -
accuracy: 0.9387 - val_loss: 22.9238 - val_accuracy: 0.0324
Epoch 409/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2079 -
accuracy: 0.9399 - val_loss: 22.9685 - val_accuracy: 0.0324
Epoch 410/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2045 -
accuracy: 0.9399 - val_loss: 22.9523 - val_accuracy: 0.0315
Epoch 411/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2025 -
accuracy: 0.9402 - val_loss: 22.9638 - val_accuracy: 0.0297
Epoch 412/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2024 -
accuracy: 0.9395 - val_loss: 22.9664 - val_accuracy: 0.0315
Epoch 413/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2017 -
accuracy: 0.9403 - val_loss: 23.0098 - val_accuracy: 0.0280
Epoch 414/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2016 -
accuracy: 0.9399 - val_loss: 22.9871 - val_accuracy: 0.0315
Epoch 415/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2010 -
accuracy: 0.9398 - val_loss: 23.0121 - val_accuracy: 0.0297
Epoch 416/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2011 -
accuracy: 0.9403 - val_loss: 23.0124 - val_accuracy: 0.0306
Epoch 417/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2008 -
accuracy: 0.9400 - val_loss: 23.0041 - val_accuracy: 0.0315
Epoch 418/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2009 -
accuracy: 0.9398 - val_loss: 23.0459 - val_accuracy: 0.0315
```

```
Epoch 419/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2002 -
accuracy: 0.9395 - val_loss: 23.0271 - val_accuracy: 0.0324
Epoch 420/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2001 -
accuracy: 0.9399 - val_loss: 23.0464 - val_accuracy: 0.0315
Epoch 421/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2008 -
accuracy: 0.9402 - val_loss: 23.0961 - val_accuracy: 0.0324
Epoch 422/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2008 -
accuracy: 0.9399 - val_loss: 23.0830 - val_accuracy: 0.0289
Epoch 423/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2004 -
accuracy: 0.9396 - val_loss: 23.1151 - val_accuracy: 0.0297
Epoch 424/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2000 -
accuracy: 0.9400 - val_loss: 23.0831 - val_accuracy: 0.0341
Epoch 425/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2004 -
accuracy: 0.9399 - val_loss: 23.1416 - val_accuracy: 0.0306
Epoch 426/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2001 -
accuracy: 0.9398 - val_loss: 23.0681 - val_accuracy: 0.0324
Epoch 427/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2005 -
accuracy: 0.9392 - val_loss: 23.0963 - val_accuracy: 0.0315
Epoch 428/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2004 -
accuracy: 0.9390 - val_loss: 23.1281 - val_accuracy: 0.0315
Epoch 429/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2929 -
accuracy: 0.9196 - val_loss: 22.7769 - val_accuracy: 0.0280
Epoch 430/500
170/170 [==============================] - 5s 29ms/step - loss: 0.4365 -
accuracy: 0.8820 - val_loss: 22.9998 - val_accuracy: 0.0262
Epoch 431/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2737 -
accuracy: 0.9248 - val_loss: 23.0886 - val_accuracy: 0.0289
Epoch 432/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2208 -
accuracy: 0.9376 - val_loss: 23.0918 - val_accuracy: 0.0271
Epoch 433/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2054 -
accuracy: 0.9400 - val_loss: 23.1787 - val_accuracy: 0.0289
Epoch 434/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2015 -
accuracy: 0.9394 - val_loss: 23.2119 - val_accuracy: 0.0280
```

```
Epoch 435/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2003 -
accuracy: 0.9409 - val_loss: 23.1920 - val_accuracy: 0.0289
Epoch 436/500
170/170 [==============================] - 5s 31ms/step - loss: 0.2000 -
accuracy: 0.9399 - val_loss: 23.2473 - val_accuracy: 0.0297
Epoch 437/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1991 -
accuracy: 0.9398 - val_loss: 23.2412 - val_accuracy: 0.0297
Epoch 438/500
170/170 [==============================] - 5s 32ms/step - loss: 0.1988 -
accuracy: 0.9395 - val_loss: 23.2494 - val_accuracy: 0.0315
Epoch 439/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1988 -
accuracy: 0.9401 - val_loss: 23.2588 - val_accuracy: 0.0297
Epoch 440/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1986 -
accuracy: 0.9399 - val_loss: 23.2906 - val_accuracy: 0.0306
Epoch 441/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1981 -
accuracy: 0.9395 - val_loss: 23.2766 - val_accuracy: 0.0306
Epoch 442/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1982 -
accuracy: 0.9391 - val_loss: 23.3034 - val_accuracy: 0.0306
Epoch 443/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1981 -
accuracy: 0.9399 - val_loss: 23.3079 - val_accuracy: 0.0315
Epoch 444/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1984 -
accuracy: 0.9401 - val_loss: 23.3213 - val_accuracy: 0.0297
Epoch 445/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1978 -
accuracy: 0.9392 - val_loss: 23.3186 - val_accuracy: 0.0306
Epoch 446/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1980 -
accuracy: 0.9393 - val_loss: 23.3415 - val_accuracy: 0.0315
Epoch 447/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1977 -
accuracy: 0.9400 - val_loss: 23.3384 - val_accuracy: 0.0297
Epoch 448/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1979 -
accuracy: 0.9400 - val_loss: 23.3566 - val_accuracy: 0.0297
Epoch 449/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1979 -
accuracy: 0.9394 - val_loss: 23.3875 - val_accuracy: 0.0306
Epoch 450/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1977 -
accuracy: 0.9398 - val_loss: 23.3778 - val_accuracy: 0.0306
```

```
Epoch 451/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1990 -
accuracy: 0.9394 - val_loss: 23.4213 - val_accuracy: 0.0306
Epoch 452/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2037 -
accuracy: 0.9394 - val_loss: 23.3532 - val_accuracy: 0.0262
Epoch 453/500
170/170 [==============================] - 5s 30ms/step - loss: 0.3293 -
accuracy: 0.9097 - val_loss: 23.3438 - val_accuracy: 0.0289
Epoch 454/500
170/170 [==============================] - 5s 31ms/step - loss: 0.3608 -
accuracy: 0.9027 - val_loss: 23.3692 - val_accuracy: 0.0306
Epoch 455/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2472 -
accuracy: 0.9317 - val_loss: 23.4213 - val_accuracy: 0.0289
Epoch 456/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2081 -
accuracy: 0.9383 - val_loss: 23.5019 - val_accuracy: 0.0297
Epoch 457/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2006 -
accuracy: 0.9402 - val_loss: 23.5134 - val_accuracy: 0.0306
Epoch 458/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1985 -
accuracy: 0.9394 - val_loss: 23.5335 - val_accuracy: 0.0271
Epoch 459/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1975 -
accuracy: 0.9395 - val_loss: 23.5414 - val_accuracy: 0.0297
Epoch 460/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1976 -
accuracy: 0.9392 - val_loss: 23.5609 - val_accuracy: 0.0297
Epoch 461/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1968 -
accuracy: 0.9398 - val_loss: 23.5662 - val_accuracy: 0.0280
Epoch 462/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1966 -
accuracy: 0.9397 - val_loss: 23.5402 - val_accuracy: 0.0289
Epoch 463/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1966 -
accuracy: 0.9391 - val_loss: 23.5659 - val_accuracy: 0.0271
Epoch 464/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1963 -
accuracy: 0.9399 - val_loss: 23.5911 - val_accuracy: 0.0280
Epoch 465/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1963 -
accuracy: 0.9397 - val_loss: 23.5996 - val_accuracy: 0.0280
Epoch 466/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1958 -
accuracy: 0.9401 - val_loss: 23.5815 - val_accuracy: 0.0289
```

```
Epoch 467/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1963 -
accuracy: 0.9398 - val_loss: 23.6248 - val_accuracy: 0.0289
Epoch 468/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1960 -
accuracy: 0.9399 - val_loss: 23.6099 - val_accuracy: 0.0315
Epoch 469/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1959 -
accuracy: 0.9395 - val_loss: 23.6391 - val_accuracy: 0.0306
Epoch 470/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1964 -
accuracy: 0.9403 - val_loss: 23.6207 - val_accuracy: 0.0297
Epoch 471/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1961 -
accuracy: 0.9392 - val_loss: 23.6446 - val_accuracy: 0.0306
Epoch 472/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1957 -
accuracy: 0.9397 - val_loss: 23.6452 - val_accuracy: 0.0289
Epoch 473/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1958 -
accuracy: 0.9403 - val_loss: 23.6914 - val_accuracy: 0.0297
Epoch 474/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1961 -
accuracy: 0.9401 - val_loss: 23.6524 - val_accuracy: 0.0289
Epoch 475/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1958 -
accuracy: 0.9396 - val_loss: 23.7008 - val_accuracy: 0.0315
Epoch 476/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1957 -
accuracy: 0.9401 - val_loss: 23.7313 - val_accuracy: 0.0306
Epoch 477/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1968 -
accuracy: 0.9394 - val_loss: 23.6858 - val_accuracy: 0.0315
Epoch 478/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1991 -
accuracy: 0.9397 - val_loss: 23.6439 - val_accuracy: 0.0297
Epoch 479/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3699 -
accuracy: 0.8964 - val_loss: 23.5230 - val_accuracy: 0.0297
Epoch 480/500
170/170 [==============================] - 5s 29ms/step - loss: 0.3815 -
accuracy: 0.8945 - val_loss: 23.4624 - val_accuracy: 0.0315
Epoch 481/500
170/170 [==============================] - 5s 29ms/step - loss: 0.2352 -
accuracy: 0.9330 - val_loss: 23.5223 - val_accuracy: 0.0289
Epoch 482/500
170/170 [==============================] - 5s 30ms/step - loss: 0.2045 -
accuracy: 0.9399 - val_loss: 23.5477 - val_accuracy: 0.0306
```

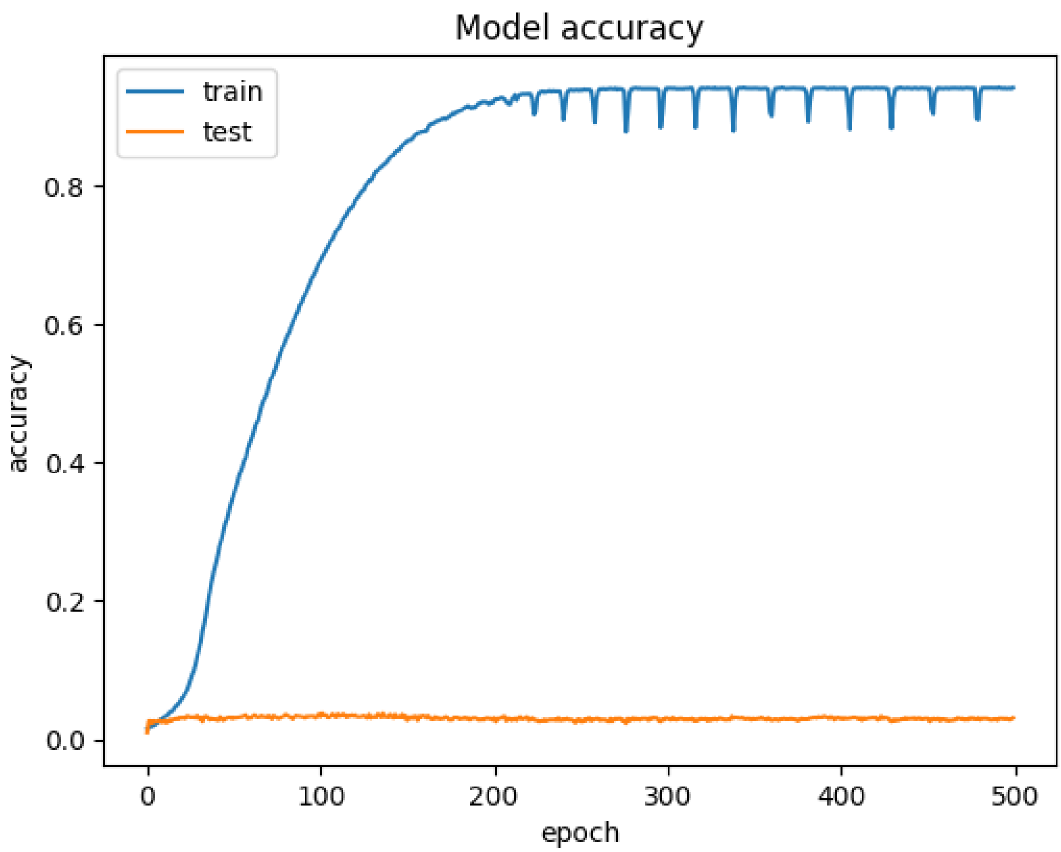
```
Epoch 483/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1984 -
accuracy: 0.9403 - val_loss: 23.5580 - val_accuracy: 0.0306
Epoch 484/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1966 -
accuracy: 0.9400 - val_loss: 23.5524 - val_accuracy: 0.0306
Epoch 485/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1958 -
accuracy: 0.9401 - val_loss: 23.5777 - val_accuracy: 0.0306
Epoch 486/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1950 -
accuracy: 0.9402 - val_loss: 23.6180 - val_accuracy: 0.0297
Epoch 487/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1950 -
accuracy: 0.9402 - val_loss: 23.6227 - val_accuracy: 0.0289
Epoch 488/500
170/170 [==============================] - 5s 29ms/step - loss: 0.1947 -
accuracy: 0.9403 - val_loss: 23.6474 - val_accuracy: 0.0289
Epoch 489/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1945 -
accuracy: 0.9399 - val_loss: 23.6490 - val_accuracy: 0.0297
Epoch 490/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1939 -
accuracy: 0.9403 - val_loss: 23.6691 - val_accuracy: 0.0297
Epoch 491/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1938 -
accuracy: 0.9396 - val_loss: 23.6729 - val_accuracy: 0.0280
Epoch 492/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1940 -
accuracy: 0.9410 - val_loss: 23.6577 - val_accuracy: 0.0306
Epoch 493/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1940 -
accuracy: 0.9398 - val_loss: 23.6673 - val_accuracy: 0.0306
Epoch 494/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1941 -
accuracy: 0.9394 - val_loss: 23.7027 - val_accuracy: 0.0306
Epoch 495/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1933 -
accuracy: 0.9400 - val_loss: 23.7206 - val_accuracy: 0.0297
Epoch 496/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1939 -
accuracy: 0.9392 - val_loss: 23.7242 - val_accuracy: 0.0306
Epoch 497/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1937 -
accuracy: 0.9395 - val_loss: 23.7191 - val_accuracy: 0.0297
Epoch 498/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1939 -
accuracy: 0.9398 - val_loss: 23.7579 - val_accuracy: 0.0289
```

```
Epoch 499/500
170/170 [==============================] - 5s 31ms/step - loss: 0.1942 -
accuracy: 0.9391 - val_loss: 23.7582 - val_accuracy: 0.0297
Epoch 500/500
170/170 [==============================] - 5s 30ms/step - loss: 0.1941 -
accuracy: 0.9404 - val_loss: 23.7793 - val_accuracy: 0.0315
```

1.13 Model Evaluation

[49]:

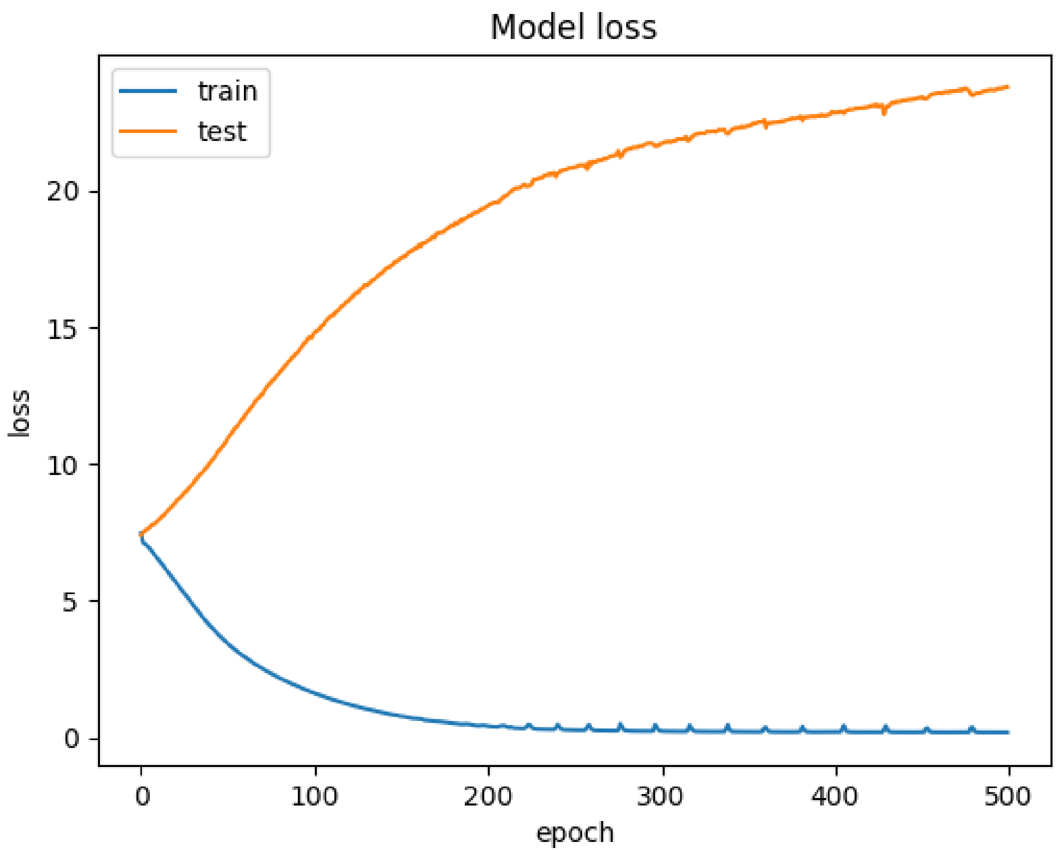
```python
plt.plot(history['accuracy'])
plt.plot(history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



The Model accuracy graph suggests that the training accuracy increases from 0 to 94% but the validation accuracy remains stagnant at around 3% over 500 epochs.

[50]:

```
plt.plot(history['loss'])
plt.plot(history['val_loss'])
plt.title('Model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```



The model loss graph suggests that the model training loss steadily decreases over time from around 7 to 0 while the validation loss steadily increases over the number of epochs from 7 to 25.

1.14 Model Prediction

[53]:

```
import time

text = "husband us find good arthritis stretching"


for i in range(10):
```

```python
# tokenize
token_text = tokenizer.texts_to_sequences([text])[0]

# padding
padded_token_text = pad_sequences([token_text], maxlen=194, padding='pre')

# predict
pos = np.argmax(model.predict(padded_token_text))

for word,index in tokenizer.word_index.items():
  if index == pos:
    text = text + " " + word
    print(text)
    time.sleep(2)
```

```
1/1 [==============================] - 0s 31ms/step
husband us find good arthritis stretching exercise
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely insane
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely insane winter
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely insane winter
highly
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely insane winter
highly recommended
1/1 [==============================] - 0s 22ms/step
husband us find good arthritis stretching exercise one completely insane winter
highly recommended peel
1/1 [==============================] - 0s 21ms/step
husband us find good arthritis stretching exercise one completely insane winter
highly recommended peel model
1/1 [==============================] - 0s 24ms/step
husband us find good arthritis stretching exercise one completely insane winter
highly recommended peel model soon
```

1.15 Conclusion for Positive Review Data

From the model performance graphs we can see that the model is able to achieve a high training accuracy over time but is not able to achieve a good validation accuracy and hence we can conclude that the model is biased and is overfitting and is not able to generalize on unseen data.

Also, while the model training loss decreases with the number of epochs, the validation loss increases consistently with time which also confirms that the model is overfitting and is not able to generalize well on unseen data.

1.16 Convert the file into pdf and html format

[54]:
```
%%shell
jupyter nbconvert --to html ///content/HW2_RNN_Positive_Reviews.ipynb
```

```
[NbConvertApp] Converting notebook ///content/HW2_RNN_Positive_Reviews.ipynb to
html
[NbConvertApp] Writing 1373179 bytes to /content/HW2_RNN_Positive_Reviews.html
```

[54]: