SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ZADANIE 3 : KONVOLUČNÉ NEURÓNOVÉ SIETE A PRENOS VEDOMOSTÍ

SEMINÁRNA PRÁCA

2022 Ladislav Rajcsányi

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ZADANIE 3 : KONVOLUČNÉ NEURÓNOVÉ SIETE A PRENOS VEDOMOSTÍ

SEMINÁRNA PRÁCA

Študijný program:	Aplikovaná informatika
Predmet:	I-SUNS – Strojové učenie a neurónové siete
Prednášajúci:	prof. Dr. Ing. Miloš Oravec
Cvičiaci:	Ing. Zuzana Bukovčiková
	Ing. Vanesa Andicsová
	Ing. Dominik Sopiak, PhD.

Bratislava 2022 **Ladislav Rajcsányi**

Obsah

Zoznam obrázkov a tabuliek

Zoznam skratiek

API Application Programming Interface

KNS Konvolučná Neurónová Sieť

ML Machine Learning

Úvod

Hlavným cieľom tohto zadania je predpovedanie kategórie daného jedla na obrázku pomocou Konvolučných Neurónových Sietí (Convolutional Neural Network), tým pádom úloha je triediaca (klasifikačná). Na vypracovanie zadania sme použili poskytnuté obrázky, ktoré sú rozdelené na trénovacie a testovacie dáta. Trénovacie dáta použijeme na natrénovanie konvolučných neurónových sietí a v prípade potreby môžeme z nich vybrať aj validačné dáta, ktoré môžeme používať na monitorovanie úspešnosti na predtým nevidených dát počas fázy trénovania.

Dáta sú uložené v JPG formáte. Trénovacie aj testovacie dáta sú rozdelené do 30 rôznych kategórii, ktoré sú nasledovné:

- apple_pie
- baby_back_ribs
- caesar_salad
- caprese_salad
- chicken_quesadilla
- chicken_wings
- chocolate_cake
- cup_cakes
- donuts
- dumplings
- french_fries
- garlic_bread
- grilled_salmon
- guacamole
- hamburger
- hot_dog

- ice_cream
- lasagna
- macaroni_and_cheese
- macarons
- onion_rings
- oysters
- pancakes
- pho
- pizza
- red_velvet_cake
- risotto
- sashimi
- spaghetti_bolognese
- waffles

1 Použité technológie

1.1 Pandas

Pandas je rýchly, výkonný, flexibilný a ľahko použiteľný open source nástroj na analýzu a manipuláciu s údajmi, postavený na programovacom jazyku Python [1].

1.2 Matplotlib

Matplotlib je komplexná knižnica na vytváranie statických, animovaných a interaktívnych vizualizácií v jazyku Python [2].

1.3 Seaborn

Seaborn je knižnica na vizualizáciu údajov v jazyku Python založená na matplotlib. Poskytuje vysokoúrovňové rozhranie na kreslenie atraktívnej a informatívnej štatistickej grafiky [3].

1.4 Scikit-Learn

- Jednoduché a efektívne nástroje na prediktívnu analýzu údajov
- Prístupné pre každého a opakovane použiteľné v rôznych kontextoch
- Postavené na NumPy, SciPy a matplotlib
- Open Source, komerčne použiteľný - licencia BSD [4]

1.5 TensorFlow

TensorFlow je komplexná open source platforma pre strojové učenie (angl.: Machine Learning - ML). Má komplexný, flexibilný ekosystém nástrojov, knižníc a komunitných zdrojov, ktorý umožňuje výskumníkom posúvať najnovšie poznatky v oblasti ML a vývojárom ľahko vytvárať a nasadzovať aplikácie využívajúce ML [5].

1.6 Keras

Keras je API určené pre ľudí, nie pre stroje. Keras sa riadi osvedčenými postupmi na zníženie kognitívnej záťaže: ponúka konzistentné a jednoduché API, minimalizuje počet činností používateľa potrebných pre bežné prípady použitia a poskytuje jasné a použiteľné chybové hlásenia. Má tiež rozsiahlu dokumentáciu a príručky pre vývojárov [6].

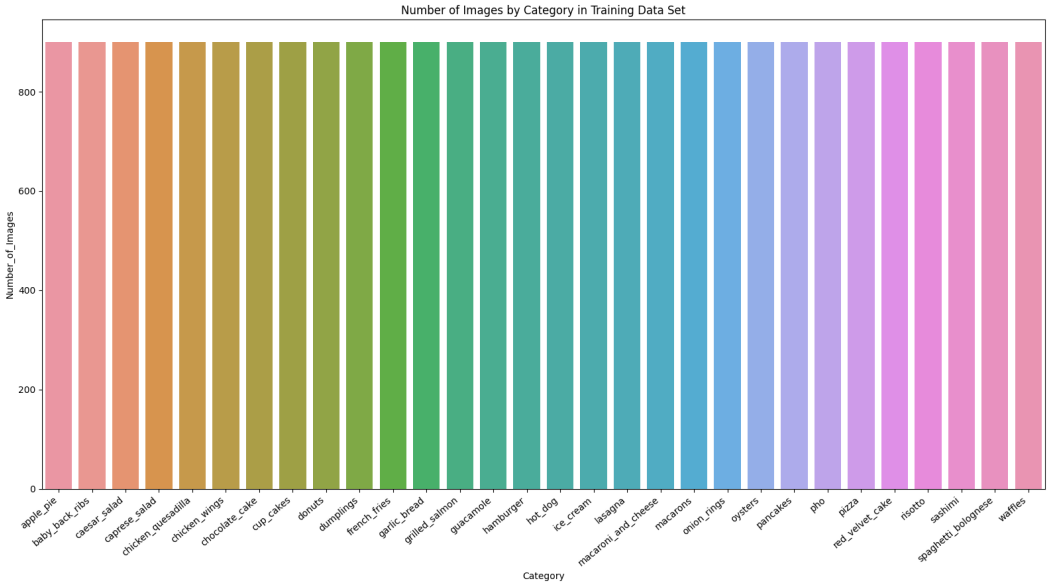
2 Implementácia

2.1 Načítanie Dát

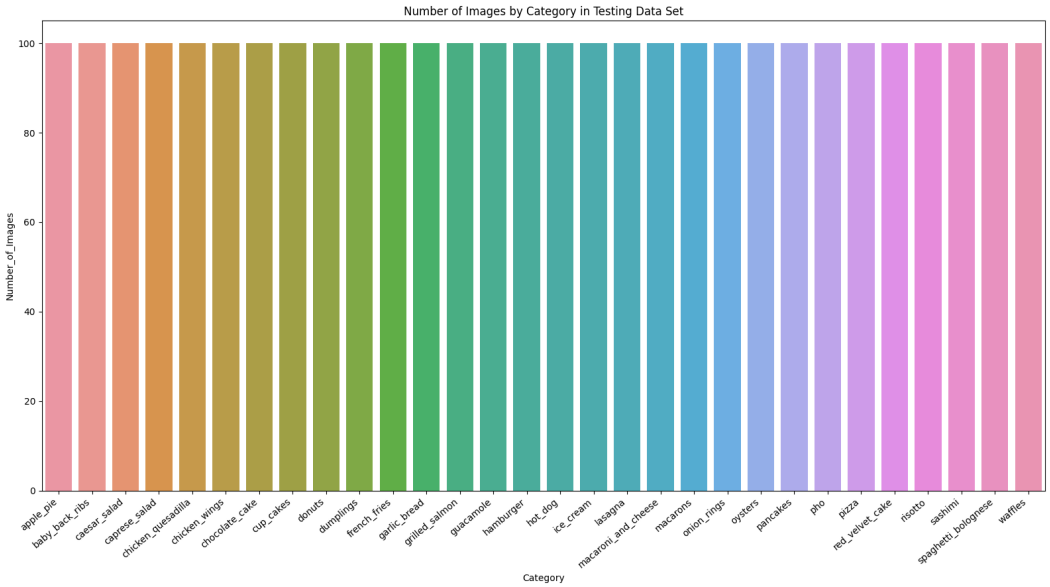
Riešenie tohto zadania začneme načítaním vopred nachystaných trénovacích aj testovacích dát, ktoré sú obrázkové dáta. Na načítanie dát sme vytvorili vlastnú funkciu (**Dataloader.load_all_data(...)**) na základe článku [7] a odpovedí [8], ktorá postupne v batchoch načíta všetky obrázky, pridelí k nim kategóriu na základe priečinku, v ktorom sa nachádzajú dané obrázky. V tomto procese normalizujeme obrázky (všetky hodnoty vynásobíme s 1/255, a tým pádom, všetky pixely v našich obrázkoch budú mať hodnoty <0.0, 1.0>) a následne ich zmenšíme, aby sme zrýchlili proces trénovania. V tomto procese sme tiež vytvorili aj validačnú množinu, ktorú budeme používať počas trénovania na skúmanie správania sa našej siete keď pracuje s vopred nevidenými dátami. V Prieskumnej Analýze Dát si overíme, či sa nám podarilo úspešne načítať všetky obrázky.

2.2 Prieskumná Analýza Dát (Exploratory Data Analysis)

Po načítaní dát nasleduje Prieskumná Analýza Dát. Začneme so zobrazením základných informácii o našich dátach. Inšpiráciu pre túto analýzu sme našli v článku [9]. Najprv sme si zobrazili ako vyzeralo rozloženie našich dát pred rozdelením. Na nižšie uvedených obrázkoch vidíte, že každá kategória má rovnaký počet prvkov. Každá kategória má v prípade trénovacích 900 a v prípade testovacích 100 prvkov.

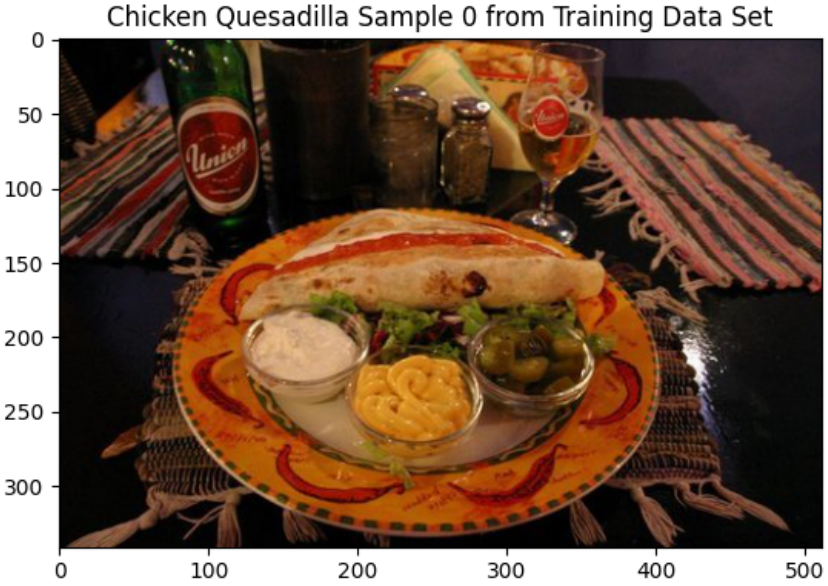


Obr. 1: Počet prvkov v kategóriach trénovacích dát

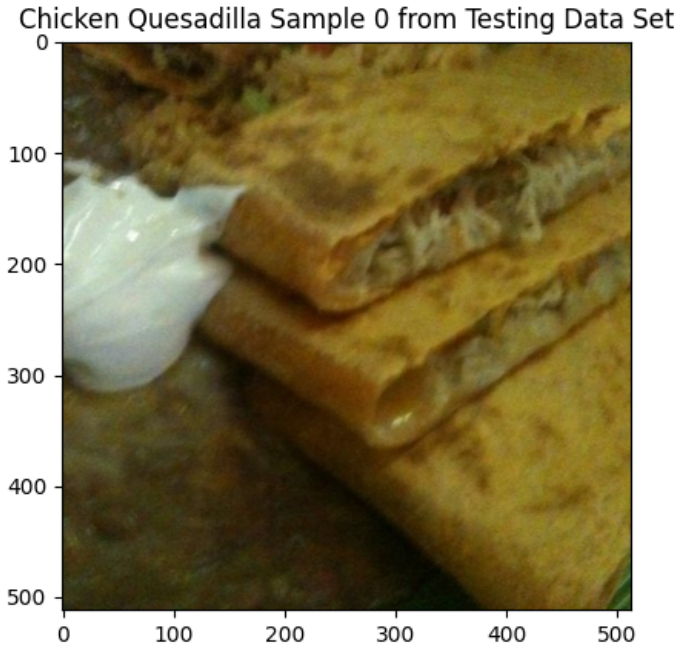


Obr. 2: Počet prvkov v kategóriách trénovacích dát

Potom zobrazili sme jeden prvok z každej kategórie z trénovacích aj testovacích dát (V dokumentácii sme nezobrazili všetky výsledky, lebo nechceli sme aby samotný dokument bol príliš dlhý, ale program Vám vygeneruje tieto výsledky). Na nižšie uvedených obrázkoch ešte vidíte pôvodnú veľkosť našich obrázkov.

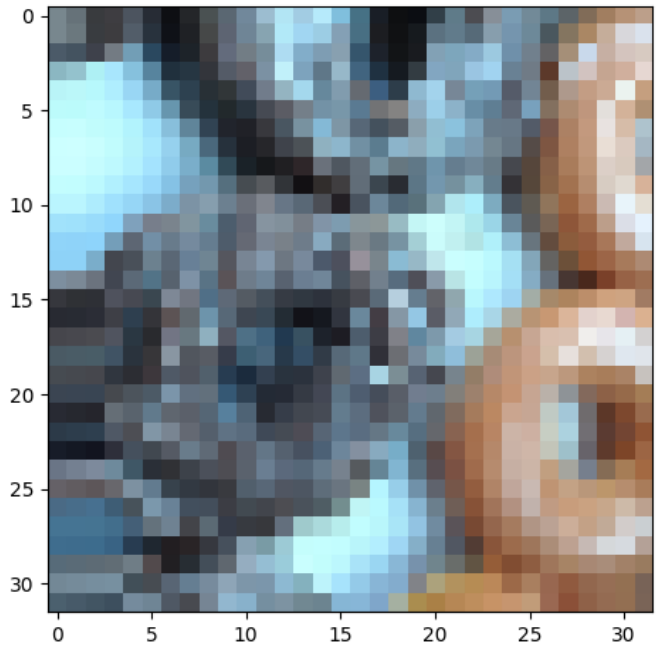


Obr. 3: Príklad z trénovacích dát



Obr. 4: Príklad z testovacích dát

Následne sme aj vyskúšali načítané dáta augmentovať (používali sme horizontal a vertical flip spolu s rotation, ale samotné trénovanie modelov sme robili na originálnych dátach), následne sme dostali podobné výsledky.



Potom zobrazili sme informáciu o veľkosti jednotlivých množín. Prvý riadok reprezentuje trénovaciu, druhá validačnú a tretia testovaciu množinu.

```
Found 20250 images belonging to 30 classes.
Found 6750 images belonging to 30 classes.
Found 3000 images belonging to 30 classes.
```

Následne sme zobrazili jednotlivé kategórie, ktoré sa nachádzajú v jednotlivých množinách.

```
Classes in the Training Data:
{
    'apple_pie': 0,
    'baby_back_ribs': 1,
    'caesar_salad': 2,
    'caprese_salad': 3,
    'chicken_quesadilla': 4,
    'chicken_wings': 5,
    'chocolate_cake': 6,
    'cup_cakes': 7,
    'donuts': 8,
    'dumplings': 9,
    'french_fries': 10,
    'garlic_bread': 11,
    'grilled_salmon': 12,
    'guacamole': 13,
    'hamburger': 14,
    'hot_dog': 15,
    'ice_cream': 16,
    'lasagna': 17,
    'macaroni_and_cheese': 18,
    'macarons': 19,
    'onion_rings': 20,
```

```
    'oysters': 21,
    'pancakes': 22,
    'pho': 23,
    'pizza': 24,
    'red_velvet_cake': 25,
    'risotto': 26,
    'sashimi': 27,
    'spaghetti_bolognese': 28,
    'waffles': 29
}

Classes in the Validation Data:
{
    'apple_pie': 0,
    'baby_back_ribs': 1,
    'caesar_salad': 2,
    'caprese_salad': 3,
    'chicken_quesadilla': 4,
    'chicken_wings': 5,
    'chocolate_cake': 6,
    'cup_cakes': 7,
    'donuts': 8,
    'dumplings': 9,
    'french_fries': 10,
    'garlic_bread': 11,
    'grilled_salmon': 12,
    'guacamole': 13,
    'hamburger': 14,
    'hot_dog': 15,
```

```
    'ice_cream': 16,
    'lasagna': 17,
    'macaroni_and_cheese': 18,
    'macarons': 19,
    'onion_rings': 20,
    'oysters': 21,
    'pancakes': 22,
    'pho': 23,
    'pizza': 24,
    'red_velvet_cake': 25,
    'risotto': 26,
    'sashimi': 27,
    'spaghetti_bolognese': 28,
    'waffles': 29
}

Classes in the Testing Data:
{
     'apple_pie': 0,
     'baby_back_ribs': 1,
     'caesar_salad': 2,
     'caprese_salad': 3,
     'chicken_quesadilla': 4,
     'chicken_wings': 5,
     'chocolate_cake': 6,
     'cup_cakes': 7,
     'donuts': 8,
     'dumplings': 9,
     'french_fries': 10,
     'garlic_bread': 11,
     'grilled_salmon': 12,
     'guacamole': 13,
     'hamburger': 14,
```

```
    'hot_dog': 15,
    'ice_cream': 16,
    'lasagna': 17,
    'macaroni_and_cheese': 18,
    'macarons': 19,
    'onion_rings': 20,
    'oysters': 21,
    'pancakes': 22,
    'pho': 23,
    'pizza': 24,
    'red_velvet_cake': 25,
    'risotto': 26,
    'sashimi': 27,
    'spaghetti_bolognese': 28,
    'waffles': 29
}
```

Na vyššie uvedených výpisoch vidíme, že každá množina má tie isté kategórie. Potom nás zaujímalo, či sme vôbec zmenili veľkosť našich obrázkov. Preto sme si zobrazili nasledujúce výpisy.

```
Image Shape in the Training Data: (32, 32, 3)
Image Shape in the Validation Data: (32, 32, 3)
Image Shape in the Testing Data: (32, 32, 3)
```

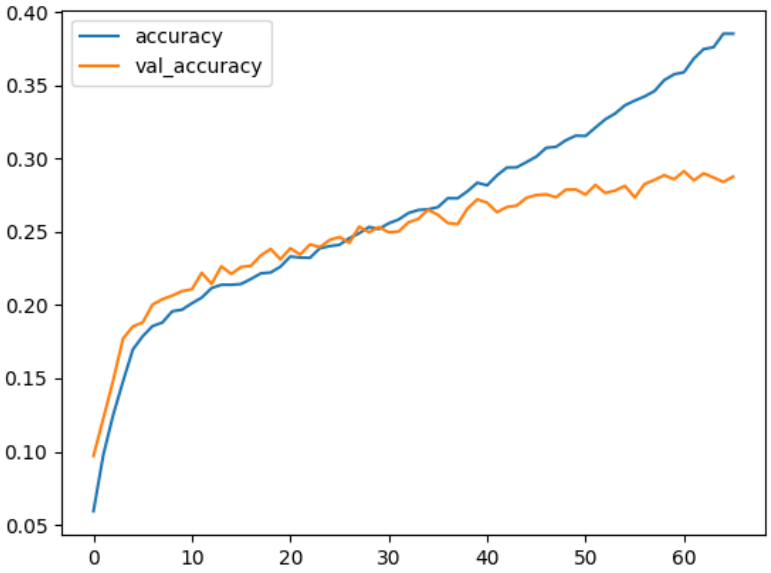
Na základe týchto grafov a výpisov sme zistili dôležité informácie o našich načítaných dátach. Úspešne sme modifikovali veľkosť našich obrázkov a rozdelili sme ich to troch množín. Keďže dáta sú už načítané a analyzované sme skončili s Prieskumnou Analýzou Dát.

2.3 Konvolučná Neurónová Sieť (Convolutional Neural Network)

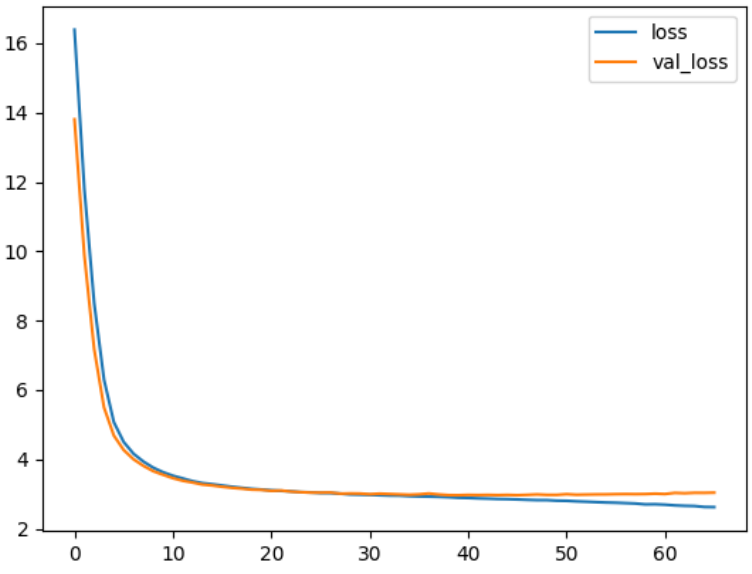
2.3.1 Najlepšia Sieť

Po Prieskumnej Analýze Dát sme natrénovali Konvolučnú Neurónovú Sieť. Na nižšie uvedenom diagrame vidíte jej štruktúru, pričom vrstva **dense** obsahuje aj L1(0.0001) regularizátor. V tejto štruktúre sme sa snažili vyhýba pretrénovaniu, preto sme nastavili **EarlyStopping** [10] s trpezlivosťou 5 epoch, ktorý nám zastavil trénovanie po 66. epoche. Sieť je minimálne pretrénovaná. Na nižšie uvedených hodnotách vidíte dosiahnuté výsledky.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 32, 32, 32)        2432

 activation (Activation)     (None, 32, 32, 32)        0

 conv2d_1 (Conv2D)           (None, 32, 32, 64)        18496

 activation_1 (Activation)   (None, 32, 32, 64)        0

 max_pooling2d (MaxPooling2D) (None, 16, 16, 64)       0

 conv2d_2 (Conv2D)           (None, 16, 16, 64)        36928

 activation_2 (Activation)   (None, 16, 16, 64)        0

 flatten (Flatten)           (None, 16384)             0

 dense (Dense)               (None, 1024)              16778240

 activation_3 (Activation)   (None, 1024)              0

 dropout (Dropout)           (None, 1024)              0

 dense_1 (Dense)             (None, 30)                30750

=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 5: Vývoj úspešnosti počas trénovania



Obr. 6: Vývoj chybovej hodnoty počas trénovania

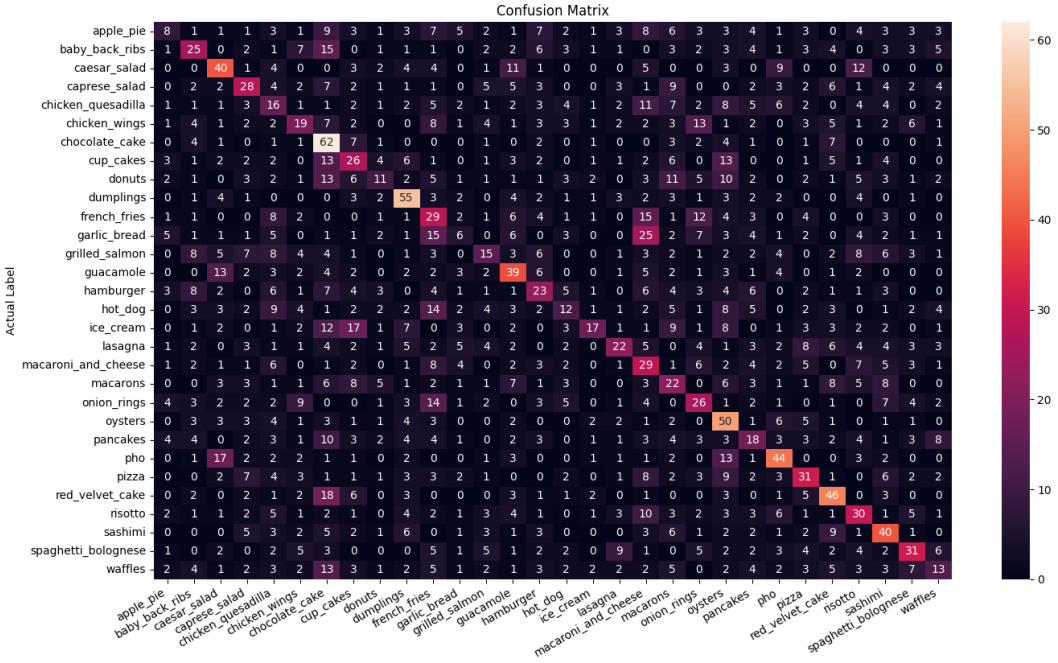
Model evaluation:

Loss: 3.034605026245117

Accuracy: 0.2776666581630707

Classification Report:

	precision	recall	f1-score	support
0	0.20	0.08	0.11	100
1	0.30	0.25	0.27	100
2	0.36	0.40	0.38	100
3	0.31	0.28	0.30	100
4	0.14	0.16	0.15	100
5	0.25	0.19	0.21	100
6	0.28	0.62	0.38	100
7	0.23	0.26	0.25	100
8	0.24	0.11	0.15	100
9	0.44	0.55	0.49	100
10	0.19	0.29	0.23	100
11	0.13	0.06	0.08	100
12	0.24	0.15	0.18	100
13	0.33	0.39	0.36	100
14	0.25	0.23	0.24	100
15	0.20	0.12	0.15	100
16	0.45	0.17	0.25	100
17	0.35	0.22	0.27	100
18	0.18	0.29	0.22	100
19	0.18	0.22	0.20	100
20	0.25	0.26	0.25	100
21	0.28	0.50	0.36	100
22	0.20	0.18	0.19	100
23	0.41	0.44	0.43	100
24	0.32	0.31	0.31	100
25	0.38	0.46	0.42	100
26	0.27	0.30	0.29	100
27	0.33	0.40	0.36	100
28	0.38	0.31	0.34	100
29	0.21	0.13	0.16	100
accuracy			0.28	3000
macro avg	0.28	0.28	0.27	3000
weighted avg	0.28	0.28	0.27	3000



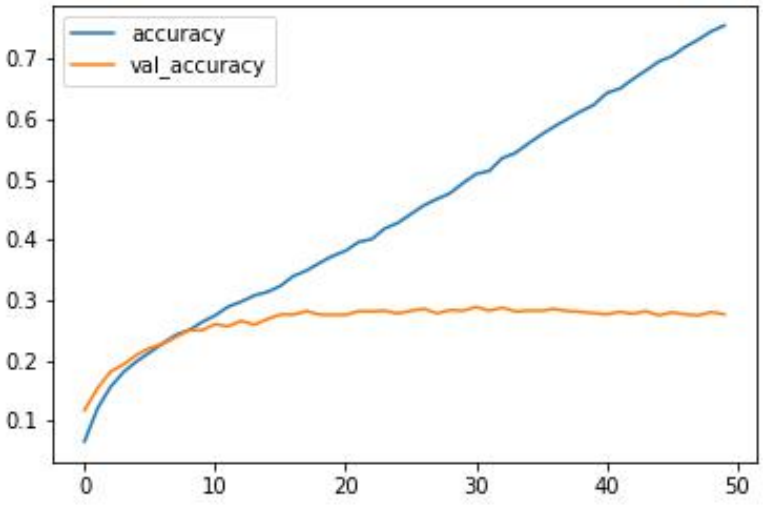
Obr. 7: Konfúzna matica

2.3.2 Pretrénovanie

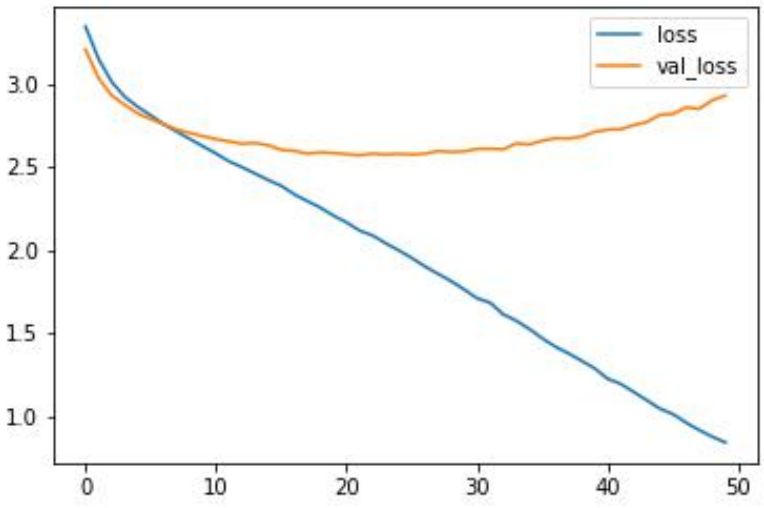
Na túto podúlohu sme používali tú istú sieť ako v prípade najlepšej siete, síce v tomto príprade sme chceli naschvál pretrénovať sieť, a kvôli tomu sme deaktivovali EarlyStopping.

```
Model: "sequential_34"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_81 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_41 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_82 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_42 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_32 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
```

```
conv2d_83 (Conv2D)          (None, 16, 16, 64)      36928
_________________________________________________________________
activation_43 (Activation) (None, 16, 16, 64)     0
_________________________________________________________________
flatten_32 (Flatten)        (None, 16384)           0
_________________________________________________________________
dense_64 (Dense)            (None, 1024)            16778240
_________________________________________________________________
activation_44 (Activation) (None, 1024)           0
_________________________________________________________________
dropout_20 (Dropout)        (None, 1024)            0
_________________________________________________________________
dense_65 (Dense)            (None, 30)              30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 8: Vývoj úspešnosti počas trénovania



Obr. 9: Vývoj chybovej hodnoty počas trénovania

Model evaluation:

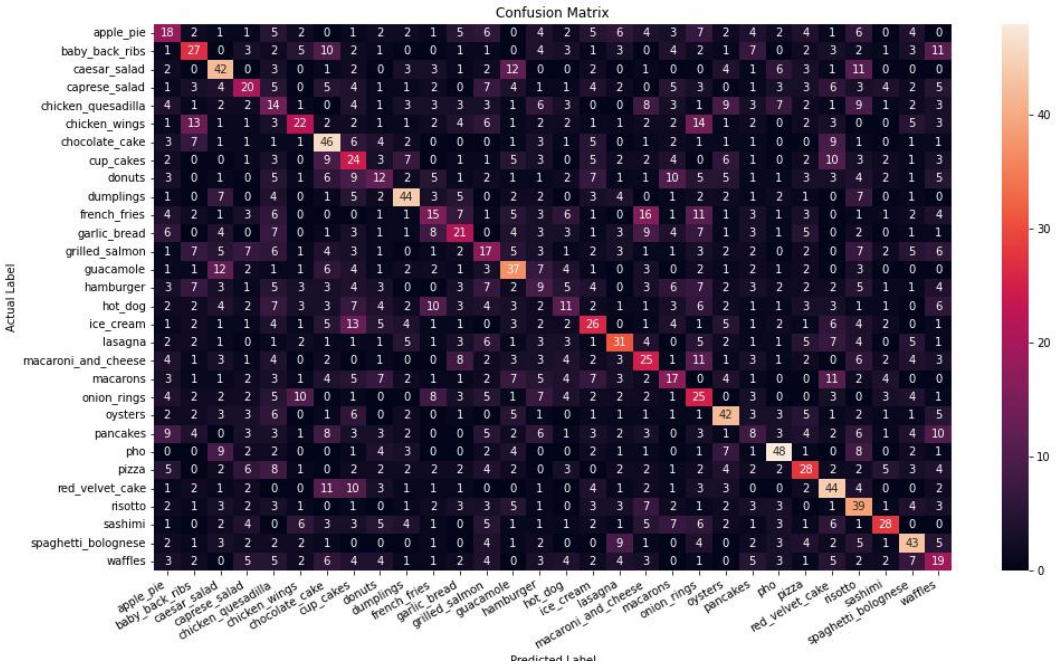
Loss: 2.9544947147369385

Accuracy: 0.26733332872390747

Classification Report:

	precision	recall	f1-score	support
0	0.20	0.18	0.19	100
1	0.29	0.27	0.28	100
2	0.35	0.42	0.38	100
3	0.25	0.20	0.22	100
4	0.11	0.14	0.13	100
5	0.33	0.22	0.26	100
6	0.33	0.46	0.38	100
7	0.18	0.24	0.21	100
8	0.16	0.12	0.14	100
9	0.45	0.44	0.44	100
10	0.20	0.15	0.17	100
11	0.25	0.21	0.23	100
12	0.17	0.17	0.17	100
13	0.31	0.37	0.34	100
14	0.10	0.09	0.10	100
15	0.16	0.11	0.13	100

16	0.26	0.26	0.26	100
17	0.35	0.31	0.33	100
18	0.23	0.25	0.24	100
19	0.20	0.17	0.18	100
20	0.19	0.25	0.21	100
21	0.37	0.42	0.39	100
22	0.11	0.08	0.09	100
23	0.48	0.48	0.48	100
24	0.30	0.28	0.29	100
25	0.33	0.44	0.38	100
26	0.26	0.39	0.31	100
27	0.43	0.28	0.34	100
28	0.41	0.43	0.42	100
29	0.18	0.19	0.18	100
accuracy			0.27	3000
macro avg	0.27	0.27	0.26	3000
weighted avg	0.27	0.27	0.26	3000



Obr. 10: Konfúzna matica

2.3.3 Regularizátory

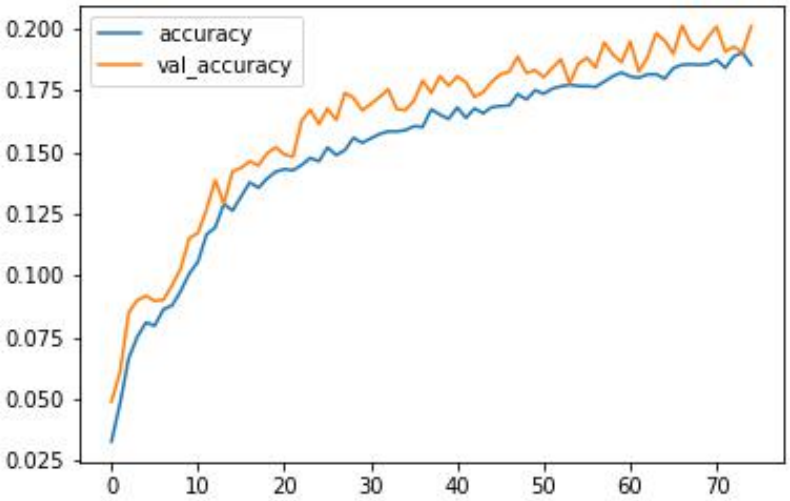
Po pretrénovaní sme vyskúšali aký vplyv majú regularizátory na pretrénovanie. Vyskúšali sme hodnoty 0.001, 0.0001 a 0.00001 pre L1 a L2 regularizátory. Dosiahli sme nasledujúce výsledky. V každom prípade sme nastavili regularizátor na Dense vrstvu, ktorá obsahuje 1024 neurónov.

2.3.3.1 L1 - 0.001

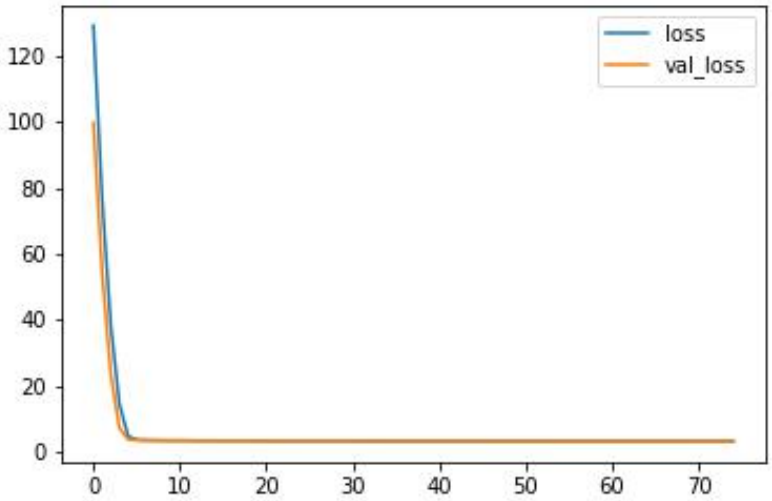
V tomto prípade L1 regularizátor zabránil pretrénovaniu.

```
Model: "sequential_30"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_69 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_25 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_70 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_26 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_28 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_71 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
activation_27 (Activation)   (None, 16, 16, 64)        0
_________________________________________________________________
flatten_28 (Flatten)         (None, 16384)             0
_________________________________________________________________
dense_56 (Dense)             (None, 1024)              16778240
_________________________________________________________________
activation_28 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_16 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_57 (Dense)             (None, 30)                30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```

```
Model evaluation:

Loss: 3.289435863494873
Accuracy: 0.18966667354106903
```



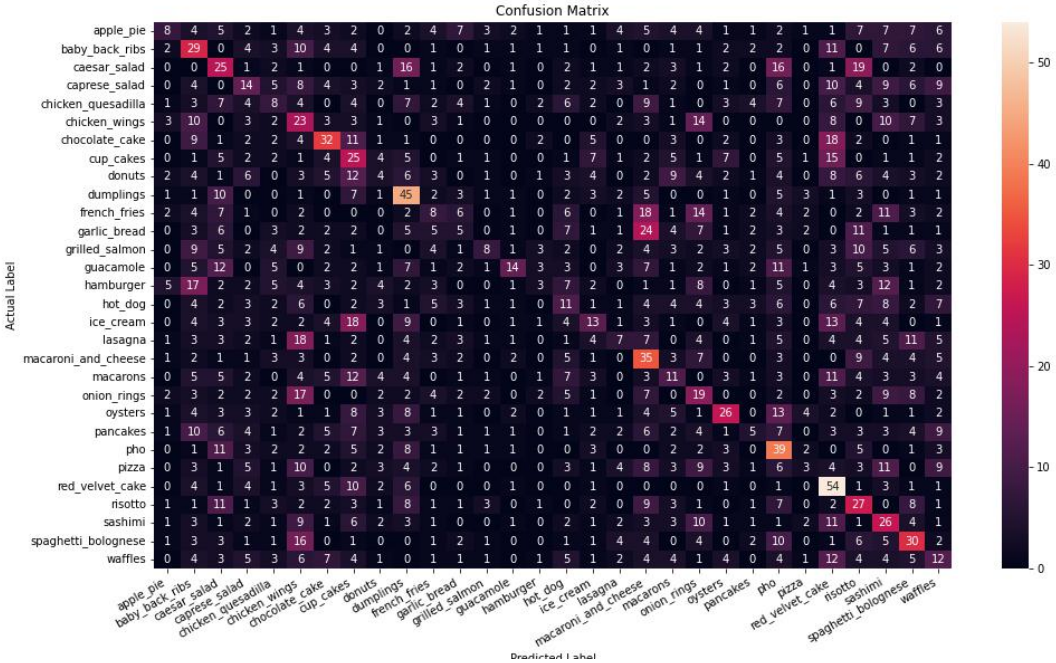
Obr. 11: Vývoj úspešnosti počas trénovania



Obr. 12: Vývoj chybovej hodnoty počas trénovania

Classification Report:

	precision	recall	f1-score	support
0	0.24	0.08	0.12	100
1	0.18	0.29	0.23	100
2	0.18	0.25	0.21	100
3	0.17	0.14	0.15	100
4	0.12	0.08	0.10	100
5	0.13	0.23	0.17	100
6	0.33	0.32	0.32	100
7	0.16	0.25	0.19	100
8	0.09	0.04	0.05	100
9	0.28	0.45	0.34	100
10	0.13	0.08	0.10	100
11	0.10	0.05	0.07	100
12	0.26	0.08	0.12	100
13	0.40	0.14	0.21	100
14	0.14	0.03	0.05	100
15	0.12	0.11	0.12	100
16	0.20	0.13	0.16	100
17	0.16	0.07	0.10	100
18	0.19	0.35	0.25	100
19	0.14	0.11	0.12	100
20	0.15	0.19	0.17	100
21	0.34	0.26	0.30	100
22	0.15	0.05	0.08	100
23	0.21	0.39	0.27	100
24	0.14	0.03	0.05	100
25	0.25	0.54	0.34	100
26	0.17	0.27	0.21	100
27	0.16	0.26	0.20	100
28	0.23	0.30	0.26	100
29	0.11	0.12	0.12	100
accuracy			0.19	3000
macro avg	0.19	0.19	0.17	3000
weighted avg	0.19	0.19	0.17	3000

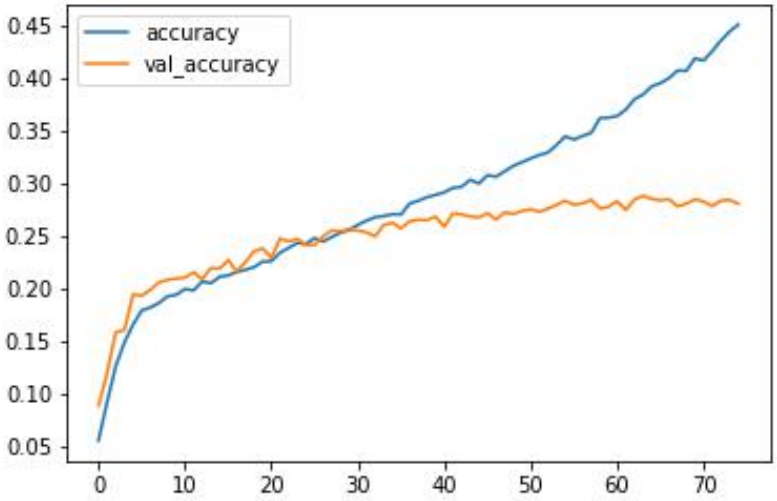


Obr. 13: Konfúzna matica

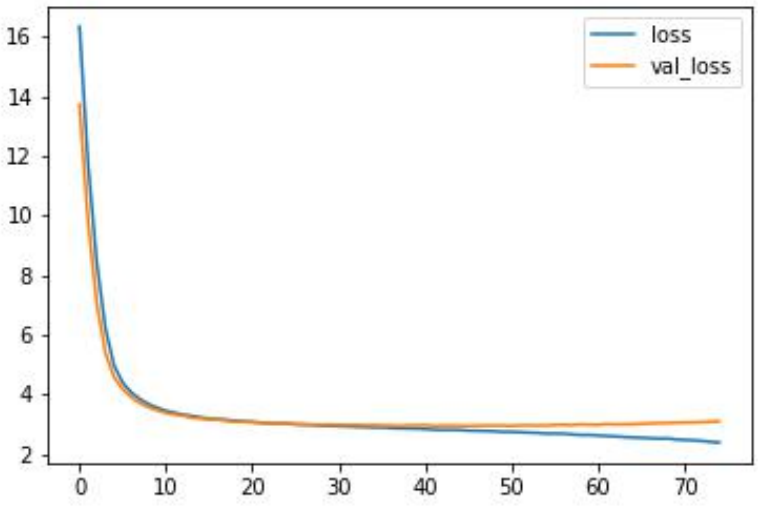
2.3.3.2 L1 - 0.0001 V tomto prípade sme vyskúšali L1 regularizátor s menšou hodnotou a v tomto prípade už nastalo pretrénovanie cca. na 30-35. epoche.

```
Model: "sequential_35"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_84 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_45 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_85 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_46 (Activation)   (None, 32, 32, 64)        0
```

```
_________________________________________________________________
max_pooling2d_33 (MaxPooling (None, 16, 16, 64)   0
_________________________________________________________________
conv2d_86 (Conv2D)           (None, 16, 16, 64)      36928
_________________________________________________________________
activation_47 (Activation) (None, 16, 16, 64)     0
_________________________________________________________________
flatten_33 (Flatten)       (None, 16384)           0
_________________________________________________________________
dense_66 (Dense)           (None, 1024)            16778240
_________________________________________________________________
activation_48 (Activation) (None, 1024)           0
_________________________________________________________________
dropout_21 (Dropout)       (None, 1024)            0
_________________________________________________________________
dense_67 (Dense)           (None, 30)              30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 14: Vývoj úspešnosti počas trénovania



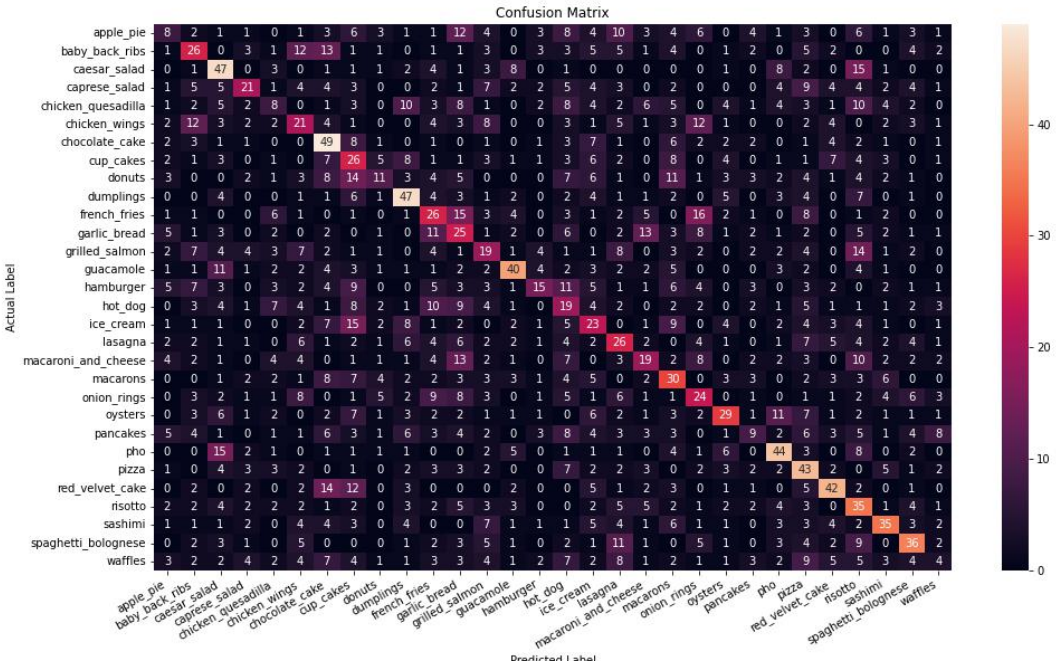
Obr. 15: Vývoj chybovej hodnoty počas trénovania

```
Model evaluation:

Loss: 3.1494674682617188
Accuracy: 0.26899999380111694
```

```
Classification Report:
```

	precision	recall	f1-score	support
0	0.15	0.08	0.10	100
1	0.27	0.26	0.27	100
2	0.35	0.47	0.40	100
3	0.36	0.21	0.26	100
4	0.14	0.08	0.10	100
5	0.21	0.21	0.21	100
6	0.32	0.49	0.38	100
7	0.17	0.26	0.21	100
8	0.24	0.11	0.15	100
9	0.40	0.47	0.43	100
10	0.22	0.26	0.24	100
11	0.18	0.25	0.21	100
12	0.19	0.19	0.19	100
13	0.48	0.40	0.43	100
14	0.33	0.15	0.21	100

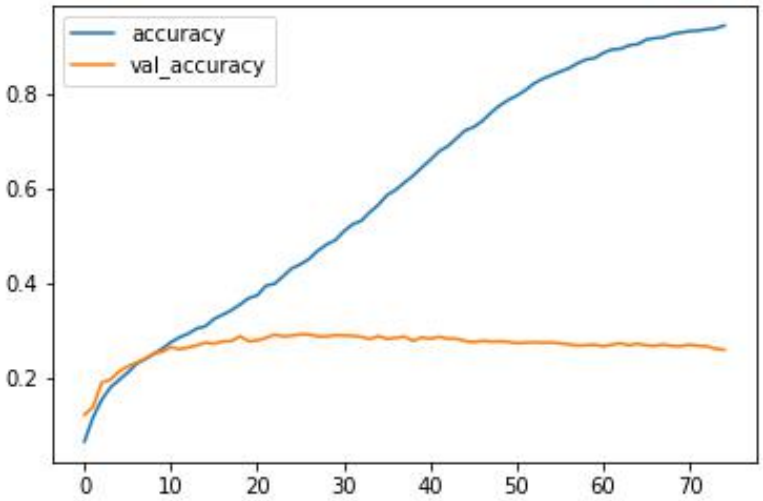
15	0.14	0.19	0.16	100
16	0.20	0.23	0.21	100
17	0.22	0.26	0.24	100
18	0.25	0.19	0.22	100
19	0.23	0.30	0.26	100
20	0.24	0.24	0.24	100
21	0.38	0.29	0.33	100
22	0.20	0.09	0.12	100
23	0.42	0.44	0.43	100
24	0.27	0.43	0.33	100
25	0.43	0.42	0.43	100
26	0.21	0.35	0.26	100
27	0.41	0.35	0.38	100
28	0.39	0.36	0.37	100
29	0.10	0.04	0.06	100
accuracy			0.27	3000
macro avg	0.27	0.27	0.26	3000
weighted avg	0.27	0.27	0.26	3000



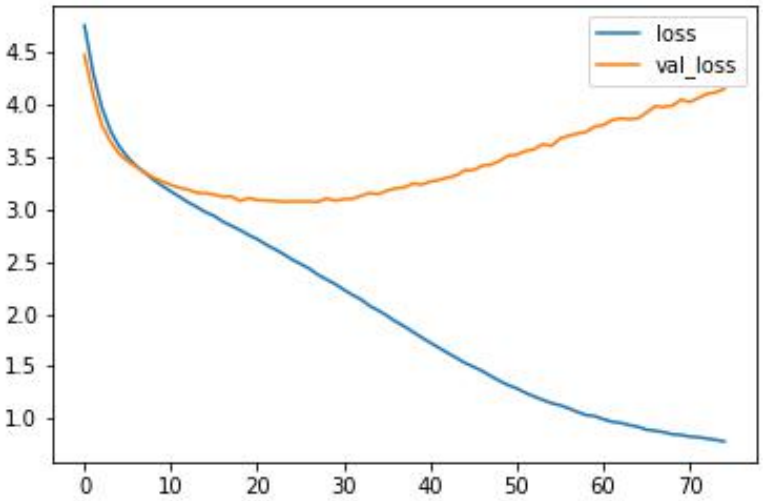
Obr. 16: Konfúzna matica

2.3.3.3 L1 - 0.00001 V tomto prípade sme vyskúšali L1 s ešte menšou hodnotou, a teraz už vôbec nezabránil pretrénovaniu, ktoré nastalo na 10 - 15. epoche.

```
Model: "sequential_36"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_87 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_49 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_88 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_50 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_34 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_89 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
activation_51 (Activation)   (None, 16, 16, 64)        0
_________________________________________________________________
flatten_34 (Flatten)         (None, 16384)             0
_________________________________________________________________
dense_68 (Dense)             (None, 1024)              16778240
_________________________________________________________________
activation_52 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_22 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_69 (Dense)             (None, 30)                30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 17: Vývoj úspešnosti počas trénovania



Obr. 18: Vývoj chybovej hodnoty počas trénovania

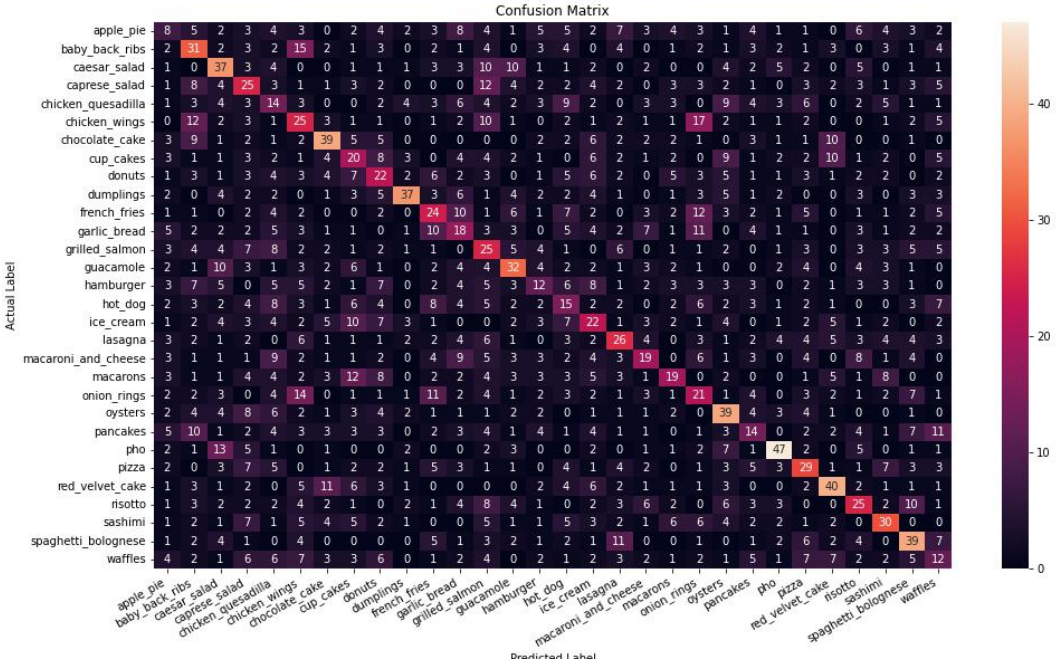
Model evaluation:

Loss: 4.162998199462891

Accuracy: 0.2553333342075348

Classification Report:

	precision	recall	f1-score	support
0	0.12	0.08	0.09	100
1	0.25	0.31	0.28	100
2	0.31	0.37	0.33	100
3	0.21	0.25	0.23	100
4	0.12	0.14	0.13	100
5	0.20	0.25	0.22	100
6	0.40	0.39	0.39	100
7	0.19	0.20	0.19	100
8	0.20	0.22	0.21	100
9	0.56	0.37	0.45	100
10	0.23	0.24	0.24	100
11	0.17	0.18	0.18	100
12	0.18	0.25	0.21	100
13	0.32	0.32	0.32	100
14	0.18	0.12	0.14	100
15	0.15	0.15	0.15	100
16	0.20	0.22	0.21	100
17	0.27	0.26	0.27	100
18	0.26	0.19	0.22	100
19	0.28	0.19	0.23	100
20	0.19	0.21	0.20	100
21	0.32	0.39	0.35	100
22	0.18	0.14	0.16	100
23	0.52	0.47	0.49	100
24	0.28	0.29	0.28	100
25	0.40	0.40	0.40	100
26	0.27	0.25	0.26	100
27	0.34	0.30	0.32	100
28	0.35	0.39	0.37	100
29	0.13	0.12	0.13	100
accuracy			0.26	3000
macro avg	0.26	0.26	0.25	3000
weighted avg	0.26	0.26	0.25	3000



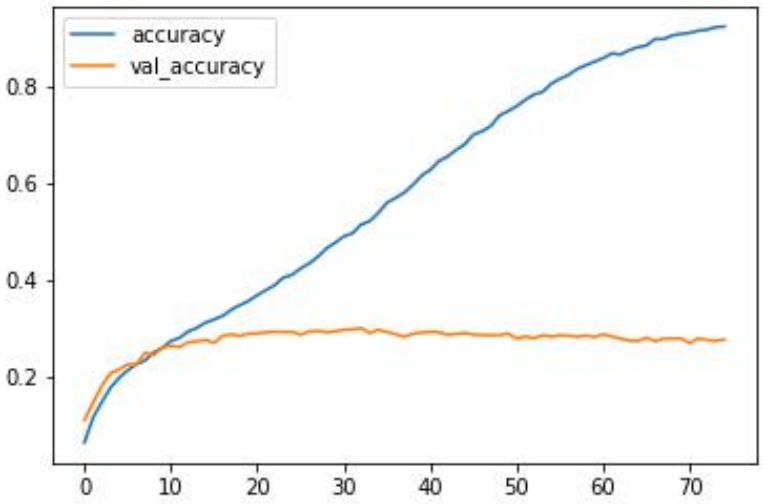
Obr. 19: Konfúzna matica

2.3.3.4 L2 - 0.001

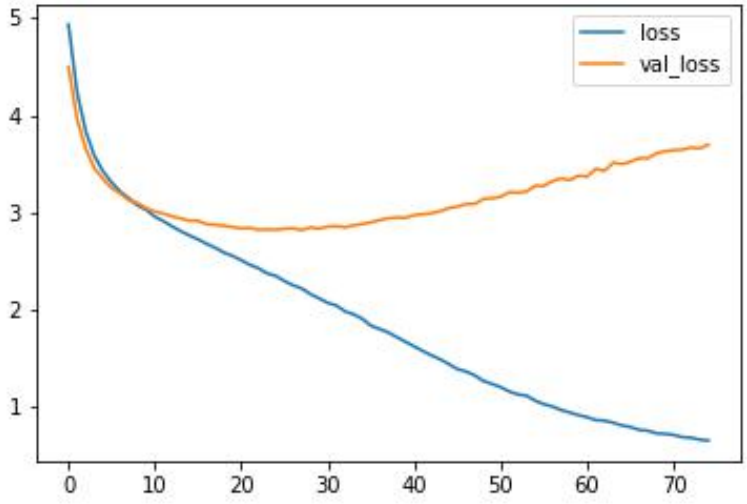
V tomto prípade už nastalo pretrénovanie cca. na 10-15. epoche.

```
Model: "sequential_37"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_90 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_53 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_91 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_54 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_35 (MaxPooling (None, 16, 16, 64)        0
```

```
_________________________________________________________________
conv2d_92 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
activation_55 (Activation)   (None, 16, 16, 64)        0
_________________________________________________________________
flatten_35 (Flatten)         (None, 16384)             0
_________________________________________________________________
dense_70 (Dense)             (None, 1024)              16778240
_________________________________________________________________
activation_56 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_23 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_71 (Dense)             (None, 30)                30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 20: Vývoj úspešnosti počas trénovania



Obr. 21: Vývoj chybovej hodnoty počas trénovania

Model evaluation:

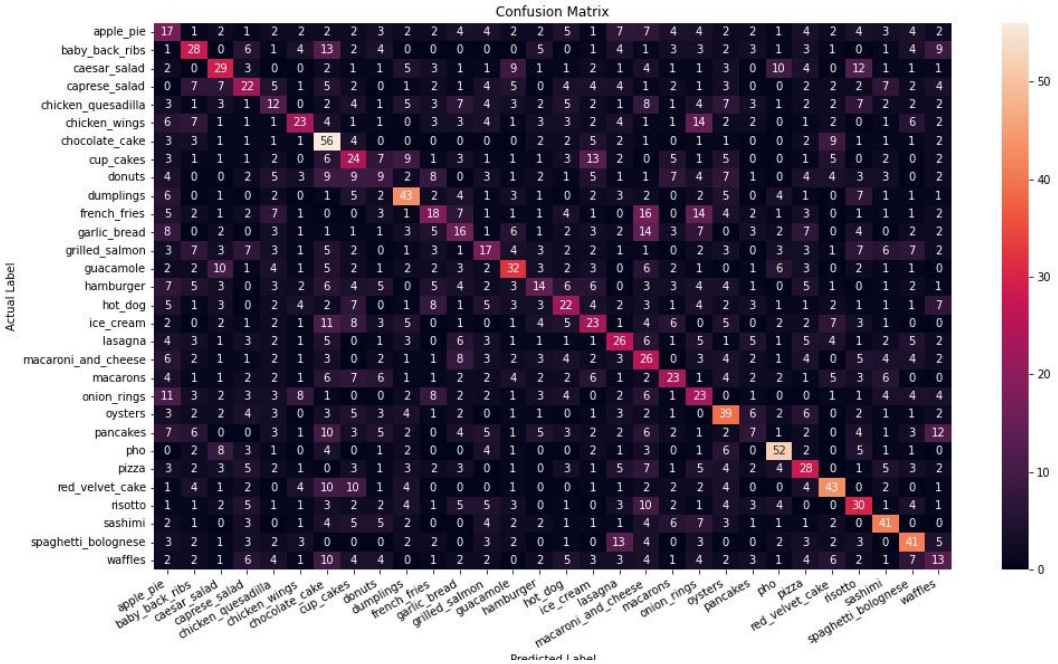
Loss: 3.7327451705932617

Accuracy: 0.265666663646698

Classification Report:

	precision	recall	f1-score	support
0	0.14	0.17	0.15	100
1	0.29	0.28	0.29	100
2	0.32	0.29	0.30	100
3	0.25	0.22	0.23	100
4	0.15	0.12	0.13	100
5	0.34	0.23	0.27	100
6	0.30	0.56	0.39	100
7	0.21	0.24	0.22	100
8	0.12	0.09	0.10	100
9	0.39	0.43	0.41	100
10	0.22	0.18	0.20	100
11	0.18	0.16	0.17	100
12	0.20	0.17	0.19	100
13	0.34	0.32	0.33	100

14	0.21	0.14	0.17	100
15	0.24	0.22	0.23	100
16	0.23	0.23	0.23	100
17	0.26	0.26	0.26	100
18	0.17	0.26	0.20	100
19	0.29	0.23	0.26	100
20	0.19	0.23	0.21	100
21	0.30	0.39	0.34	100
22	0.13	0.07	0.09	100
23	0.50	0.52	0.51	100
24	0.26	0.28	0.27	100
25	0.43	0.43	0.43	100
26	0.27	0.30	0.28	100
27	0.40	0.41	0.41	100
28	0.38	0.41	0.39	100
29	0.15	0.13	0.14	100
accuracy			0.27	3000
macro avg	0.26	0.27	0.26	3000
weighted avg	0.26	0.27	0.26	3000

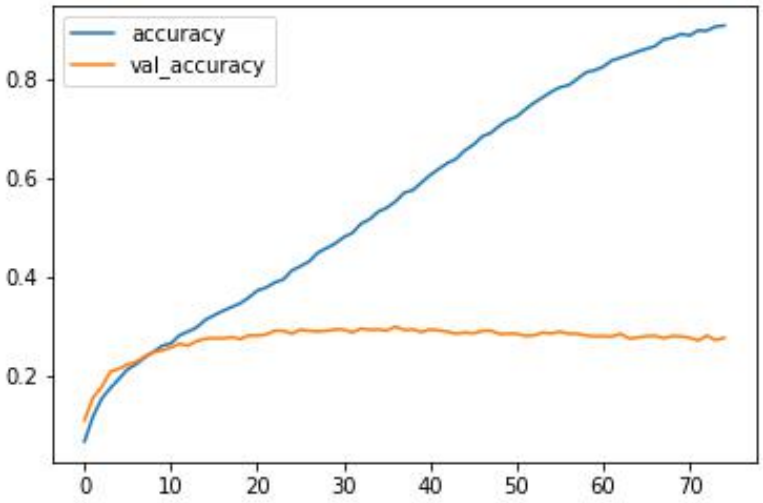


Obr. 22: Konfúzna matica

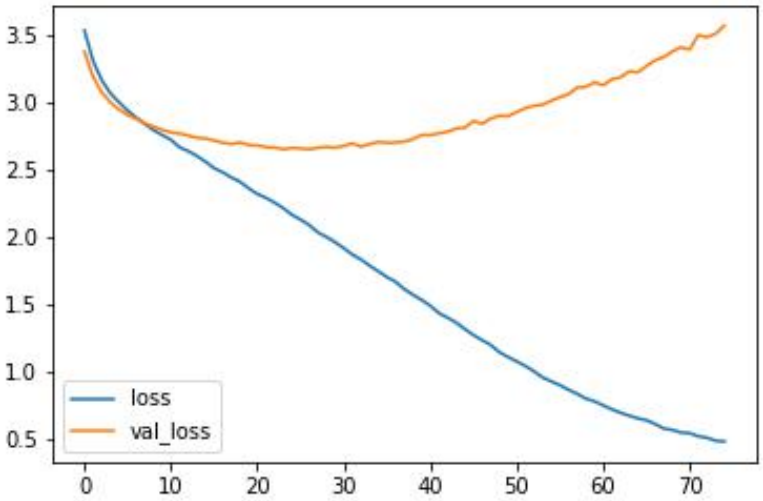
2.3.3.5 L2 - 0.0001

V tomto prípade už nastalo pretrénovanie cca. na 10-15. epoche.

```
Model: "sequential_38"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_93 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_57 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_94 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_58 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_36 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_95 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
activation_59 (Activation)   (None, 16, 16, 64)        0
_________________________________________________________________
flatten_36 (Flatten)         (None, 16384)             0
_________________________________________________________________
dense_72 (Dense)             (None, 1024)              16778240
_________________________________________________________________
activation_60 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_24 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_73 (Dense)             (None, 30)                30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 23: Vývoj úspešnosti počas trénovania



Obr. 24: Vývoj chybovej hodnoty počas trénovania

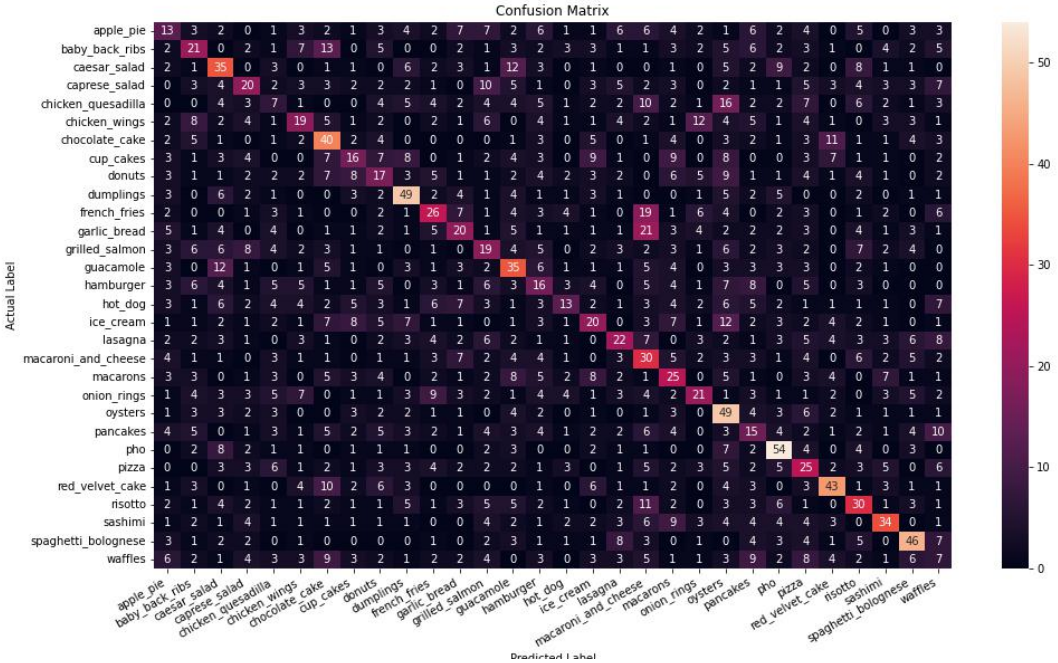
Model evaluation:

Loss: 3.6375985145568848

Accuracy: 0.26233333349227905

Classification Report:

	precision	recall	f1-score	support
0	0.17	0.13	0.15	100
1	0.24	0.21	0.22	100
2	0.29	0.35	0.32	100
3	0.26	0.20	0.23	100
4	0.10	0.07	0.08	100
5	0.25	0.19	0.22	100
6	0.30	0.40	0.34	100
7	0.23	0.16	0.19	100
8	0.18	0.17	0.18	100
9	0.42	0.49	0.45	100
10	0.29	0.26	0.27	100
11	0.24	0.20	0.22	100
12	0.19	0.19	0.19	100
13	0.28	0.35	0.31	100
14	0.16	0.16	0.16	100
15	0.27	0.13	0.18	100
16	0.23	0.20	0.21	100
17	0.28	0.22	0.25	100
18	0.19	0.30	0.23	100
19	0.22	0.25	0.23	100
20	0.29	0.21	0.24	100
21	0.26	0.49	0.34	100
22	0.15	0.15	0.15	100
23	0.43	0.54	0.48	100
24	0.20	0.25	0.22	100
25	0.45	0.43	0.44	100
26	0.28	0.30	0.29	100
27	0.40	0.34	0.37	100
28	0.43	0.46	0.45	100
29	0.08	0.07	0.07	100
accuracy			0.26	3000
macro avg	0.26	0.26	0.26	3000
weighted avg	0.26	0.26	0.26	3000

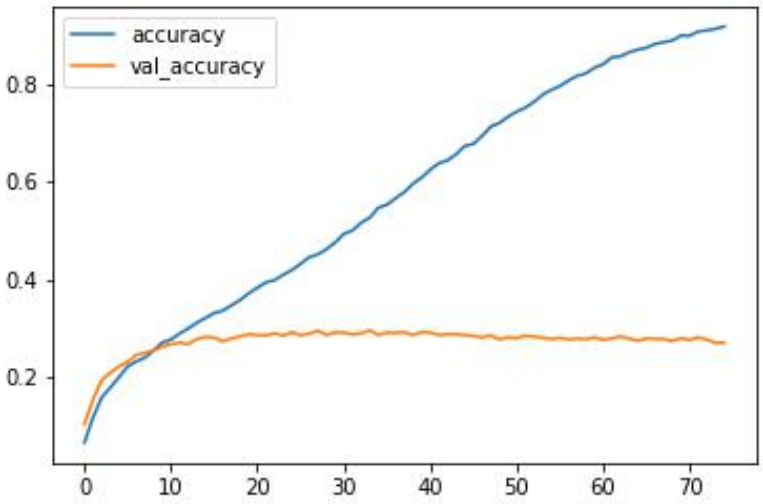


Obr. 25: Konfúzna matica

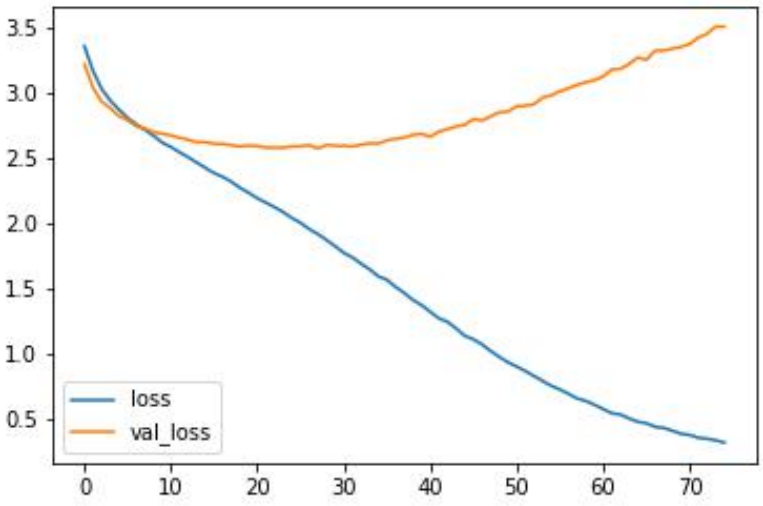
2.3.3.6 L2 - 0.00001 V tomto prípade už nastalo pretrénovanie cca. na 10-15. epoche.

```
Model: "sequential_39"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_96 (Conv2D)           (None, 32, 32, 32)        2432
_________________________________________________________________
activation_61 (Activation)   (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_97 (Conv2D)           (None, 32, 32, 64)        18496
_________________________________________________________________
activation_62 (Activation)   (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_37 (MaxPooling (None, 16, 16, 64)        0
```

```
_________________________________________________________________
conv2d_98 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
activation_63 (Activation)   (None, 16, 16, 64)        0
_________________________________________________________________
flatten_37 (Flatten)         (None, 16384)             0
_________________________________________________________________
dense_74 (Dense)             (None, 1024)              16778240
_________________________________________________________________
activation_64 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_25 (Dropout)         (None, 1024)              0
_________________________________________________________________
dense_75 (Dense)             (None, 30)                30750
=================================================================
Total params: 16,866,846
Trainable params: 16,866,846
Non-trainable params: 0
_________________________________________________________________
```



Obr. 26: Vývoj úspešnosti počas trénovania



Obr. 27: Vývoj chybovej hodnoty počas trénovania

Model evaluation:

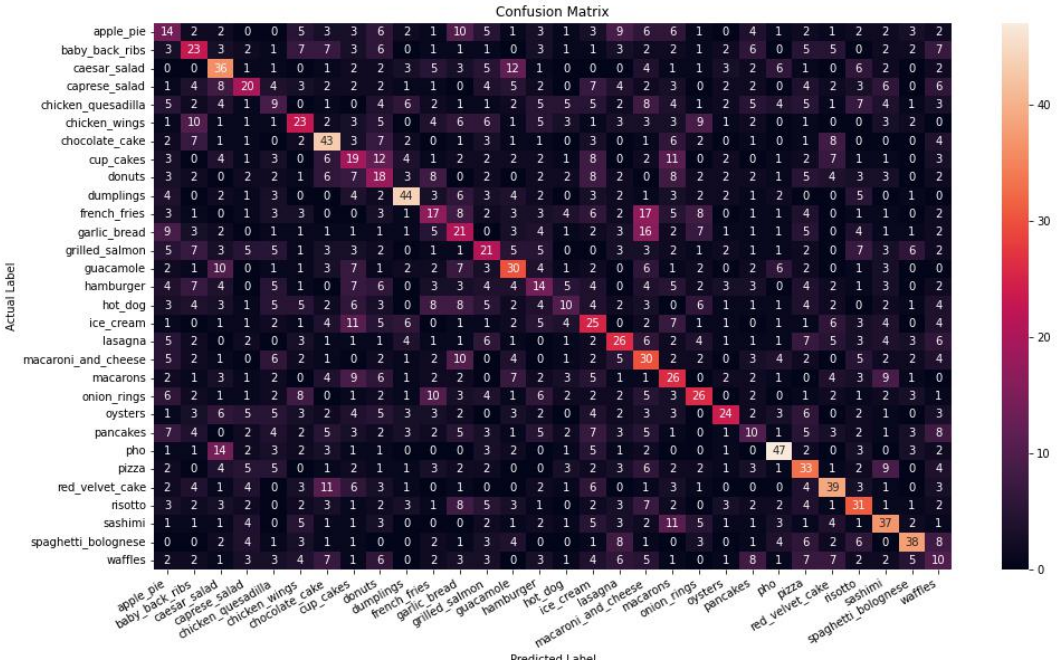
Loss: 3.528132915496826

Accuracy: 0.25466665625572205

Classification Report:

	precision	recall	f1-score	support
0	0.14	0.14	0.14	100
1	0.24	0.23	0.23	100
2	0.30	0.36	0.33	100
3	0.27	0.20	0.23	100
4	0.12	0.09	0.10	100
5	0.25	0.23	0.24	100
6	0.35	0.43	0.38	100
7	0.17	0.19	0.18	100
8	0.15	0.18	0.16	100
9	0.47	0.44	0.46	100
10	0.19	0.17	0.18	100
11	0.18	0.21	0.19	100
12	0.21	0.21	0.21	100
13	0.29	0.30	0.29	100

14	0.16	0.14	0.15	100
15	0.19	0.10	0.13	100
16	0.19	0.25	0.22	100
17	0.27	0.26	0.26	100
18	0.19	0.30	0.24	100
19	0.21	0.26	0.23	100
20	0.29	0.26	0.28	100
21	0.41	0.24	0.30	100
22	0.14	0.10	0.12	100
23	0.50	0.47	0.48	100
24	0.26	0.33	0.29	100
25	0.37	0.39	0.38	100
26	0.29	0.31	0.30	100
27	0.34	0.37	0.35	100
28	0.49	0.38	0.43	100
29	0.10	0.10	0.10	100
accuracy			0.25	3000
macro avg	0.26	0.25	0.25	3000
weighted avg	0.26	0.25	0.25	3000



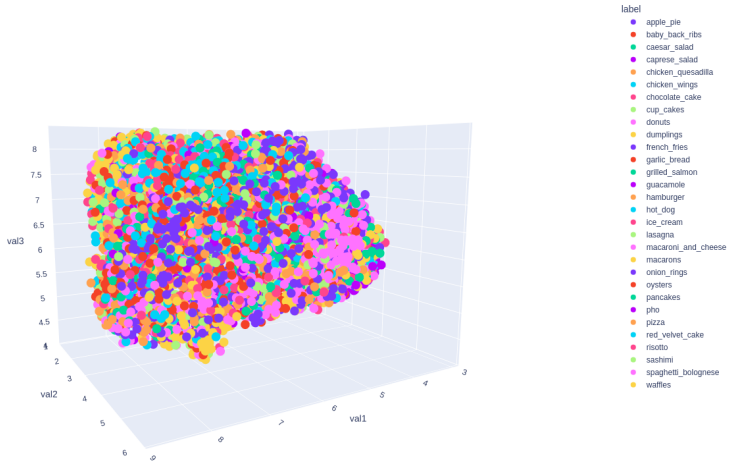
Obr. 28: Konfúzna matica

2.3.4 Predtrénovaná Sieť

Na túto podúlohu sme si vybrali **model VGG16** [11]. Používali sme **váhy ImageNet** [12]. Pomocou tejto siete sme boli schopní zakódovať a vyexportovať príznaky a kategórie do CSV súborov. Po zakódovaní a exporte sme dáta načítali do DataFrameov, a pomocou **UMAP**y [13] sme boli schopní zredukovať dimenziu príznakov. Používali sme nasledujúce nastavenia.

```
transformer = umap.UMAP(
                    n_neighbors=50,
                    random_state=420,
                    n_components=3
                ).fit(encoded_training_data.drop('label', axis=1))
```

Počet komponentov sme si vybrali na základe toho, že sme chceli zobraziť Scatter Plot v 3D, na čo potrebujeme minimálne tri parametre. Po modifikácii sme si zobrazili samotný 3D Scatter Plot (čo Vám vygeneruje program a uloží do priečinku **output/plots/3d_scatter/**.) Tieto dáta sme zobrazili pre validačnú množinu.



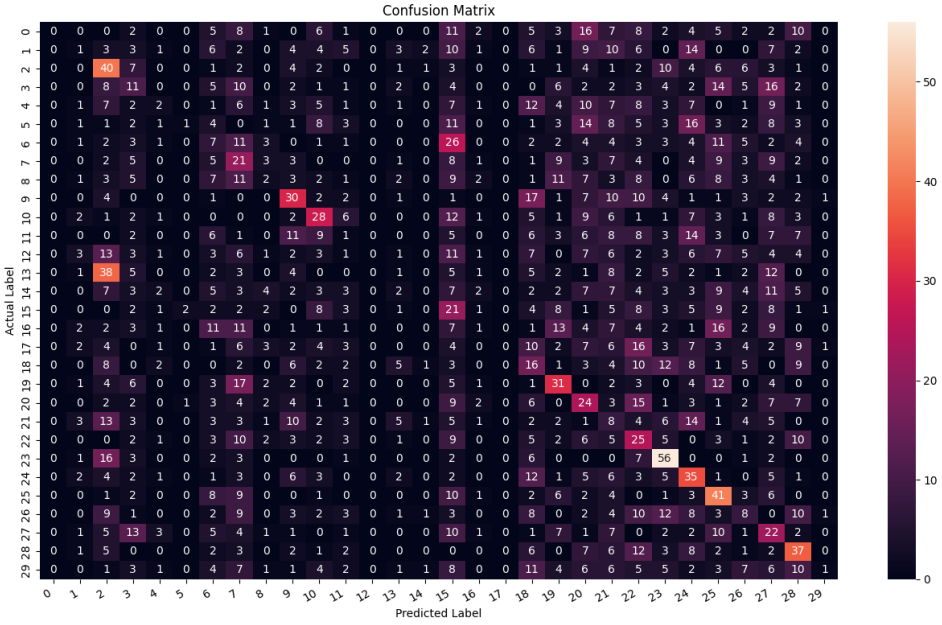
Obr. 29: 3D Scatter Plot pre Validačné Dáta

2.4 Iné Klasifikátory

V tejto podúlohe sme vyskúšali Stroje s Podpornými Vektormi (z predchádzajúceho zadania) na túto klasifikačnú úlohu. Dosiahli sme slabšie výsledky ako v prípade konvolučných neurónových sietí.

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	100
1	0.04	0.01	0.02	100
2	0.20	0.40	0.26	100
3	0.11	0.11	0.11	100
4	0.10	0.02	0.03	100
5	0.25	0.01	0.02	100
6	0.06	0.07	0.07	100
7	0.12	0.21	0.15	100
8	0.07	0.02	0.03	100
9	0.27	0.30	0.28	100
10	0.27	0.28	0.27	100
11	0.02	0.01	0.01	100
12	0.00	0.00	0.00	100
13	0.03	0.01	0.02	100
14	0.00	0.00	0.00	100
15	0.09	0.21	0.13	100
16	0.05	0.01	0.02	100
17	0.00	0.00	0.00	100
18	0.10	0.16	0.12	100
19	0.25	0.31	0.27	100
20	0.14	0.24	0.17	100
21	0.05	0.08	0.06	100
22	0.13	0.25	0.17	100
23	0.35	0.56	0.43	100
24	0.18	0.35	0.24	100
25	0.22	0.41	0.29	100
26	0.10	0.08	0.09	100
27	0.12	0.22	0.15	100
28	0.26	0.37	0.30	100
29	0.20	0.01	0.02	100
accuracy			0.16	3000
macro avg	0.13	0.16	0.12	3000
weighted avg	0.13	0.16	0.12	3000



Obr. 30: Konfúzna matica

Zoznam použitej literatúry

1. *pandas* [online] [cit. 2021-10-19]. Dostupné z : https://pandas.pydata.org/.
2. *Matplotlib* [online] [cit. 2021-10-19]. Dostupné z : https://matplotlib.org/.
3. *Seaborn - Statistical Data Visualization* [online] [cit. 2021-10-19]. Dostupné z : https://seaborn.pydata.org/.
4. *scikit-learn* [online] [cit. 2021-10-19]. Dostupné z : https://scikit-learn.org/stable/.
5. *Tensorflow* [online] [cit. 2021-10-19]. Dostupné z : https://www.tensorflow.org/.
6. *Keras - Simple. Flexible. Powerful.* Dostupné tiež z: https://keras.io/.
7. INFOARYAN. *CNN - Image Data Pre-processing with Generators* [online]. 2020-07-16 [cit. 2021-12-05]. Dostupné z : https://www.geeksforgeeks.org/cnn-image-data-pre-processing-with-generators/.
8. EZCHX a POTDAR, Keyur. *Keras split train test set when using ImageDataGenerator* [online]. 2019-06-06 [cit. 2021-12-05]. Dostupné z : https://stackoverflow.com/a/52372042.
9. RAUSCH, Dana. *EDA for Image Classification* [online]. Geek Culture, 2021-04-11 [cit. 2021-12-05]. Dostupné z : https://medium.com/geekculture/eda-for-image-classification-dcada9f2567a.
10. *Keras Documentation: EarlyStopping* [online] [cit. 2021-10-26]. Dostupné z : https://keras.io/api/callbacks/early_stopping/.
11. *VGG16 - Convolutional Network for Classification and Detection* [online]. 2018-11-20 [cit. 2021-12-08]. Dostupné z : https://neurohive.io/en/popular-networks/vgg16/.
12. *ImageNet* [online] [cit. 2021-12-08]. Dostupné z : https://image-net.org/.
13. *Uniform Manifold Approximation and Projection for Dimension Reduction* [online] [cit. 2021-12-08]. Dostupné z : https://umap-learn.readthedocs.io/en/latest/.
14. KARAGIANNAKOS, Sergios. *Best practices to write Deep learning code: Project structure, OOP, type checking and documentation* [online]. Sergios Karagiannakos, 2020-06-17 [cit. 2021-10-26]. Dostupné z : https://theaisummer.com/best-practices-deep-learning-code/.

Prílohy

A Štruktúra projektu

Inšpiráciu pre projektovú štruktúru sme našli na webovej stránke AI Summer [14].

data

· Dátové súbory

/encoded_data

· Na základe príznakov zakódované dáta

/test

· Testovacie dáta

/train

· Trénovacie dáta

dataloader

· Čítač dát

/dataloader.py

· Čítač dát

executor

· Spúšťač

/convolutional_neural_network_project.py

· Spúšťač zadania

models

· Modely Strojového Učenia

checkpoints

· Checkpointy trénovania

/convolutional_neural_network.py

· Konvolučná Neurónová Sieť

/support_vector_machine.py

· Stroj s Podpornými Vektormi

ops

· Operácie

/plotter.py

· Vykresľovač grafov

output

· Výstupy

/plots

· Grafy

/3d_scatter

· 3D Scatter Ploty

/bar_plots

· Stĺpcové Grafy

/confusion_matrices

· Konfúzne matice

/line_graphs

· Čiarové Grafy

utils

· Utilitné funkcie

/setup.py

· Setup metódy

/I-SUNS_-_Convolutional_Neural_Network_Project.pdf

· Dokumentácia - tento dokument

/main.py

· Hlavný program

/Convolutional_Neural_Network_Project

· Bash Script

/Convolutional_Neural_Network_Project.ps1

· PowerShell Script

/requirements.txt

· Zoznam požiadaných balíčkov

B Používateľská príručka

V tejto časti práce prejdeme spôsoby, ktoré nám umožňujú spúšťať túto aplikáciu.

Ak ešte nie je aktivovaný virtuálny priestor, aktivujte ho, nainštalujte potrebné balíčky a spustite main.py skript.