VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

KLASIFIKACE HISTORICKÝCH DOKUMENTŮ POMOCÍ HLUBOKÝCH NEURONOVÝCH SÍTÍ

DEEP NEURAL NETWORKS FOR HISTORICAL DOCUMENT CLASSIFICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — **Bc. BETTINA PINKEOVÁ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MARTIN KIŠŠ**
SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání diplomové práce

148667

Ústav:	Ústav počítačové grafiky a multimédií (UPGM)
Studentka:	**Pinkeová Bettina, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Strojové učení
Název:	**Klasifikace historických dokumentů pomocí hlubokých neuronových sítí**
Kategorie:	Zpracování obrazu
Akademický rok:	2022/23

Zadání:

1. Prostudujte základy konvolučních neuronových sítí a klasifikace obrazu.
2. Vytvořte si přehled o současných metodách klasifikace historických dokumentů pomocí neuronových sítí.
3. Vyberte nebo navrhněte metodu aplikovatelnou na klasifikaci historických dokumentů.
4. Obstarejte si databázi vhodnou pro experimenty.
5. Implementujte navrženou metodu a proveďte experimenty nad datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

- SEURET, Mathias, et al. ICDAR 2021 Competition on Historical Document Classification. In: *International Conference on Document Analysis and Recognition*. Springer, Cham, 2021. p. 618-634.
- KIŠŠ, Martin, et al. Importance of Textlines in Historical Document Classification. In: *International Workshop on Document Analysis Systems*. Springer, Cham, 2022. p. 158-170.

Při obhajobě semestrální části projektu je požadováno:

- Splnění prvních třech bodů zadání.
- Rozpracovaný čtvrtý a pátý bod zadání.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce:	**Kišš Martin, Ing.**
Vedoucí ústavu:	Černocký Jan, prof. Dr. Ing.
Datum zadání:	1.11.2022
Termín pro odevzdání:	17.5.2023
Datum schválení:	31.10.2022

Abstrakt

Cieľom tejto práce je vytvoriť systém na klasifikáciu historických dokumentov. Ide konkrétne o klasifikáciu dokumentov podľa miesta vzniku. V práci je navrhnutých niekoľko systémov na riešenie tohto problému. Prvý navrhnutý a realizovaný systém je založený na konvolučnej neurónovej sieti s mechanizmom *self-attention*, namiesto vrstvy združovania podľa priemeru. Ďalší systém vychádza z modelu BEiT, ktorý je postavený na vizuálnom transformery. Model BEiT sa predtrénoval na úlohu modelovanie maskovaných obrázkov a následne dotrénoval na danú klasifikačnú úlohu. Systém založený na konvolučnej neurónovej sieti dosiahol presnosť 81.6% a systém založený na modelovaní maskovaných obrázkov dosiahol prenosť 82.9%. Systémy realizované v tejto práci prevýšili úspešnosťou zúčastnených systémov na konferencie ICDAR 2021.

Abstract

The aim of this work is to create a system for historical documents classification . The task is specifically about classification of documents according to the place of origin. Several systems are proposed for solving this problem, in the work. The first designed and implemented system is based on a convolutional neural network with a *self-attention* mechanism instead of an average pooling layer. Another system is based on the BEiT model, which is built on a visual transformer. The BEiT model was pretrained on the task of masked image modelling and subsequently trained on the given classification task. The system based on convolutional neural network achieved an accuracy of 81.6% and the system based on masked image modelling achieved an accuracy of 82.9%. The systems implemented in this work, surpassed the systems participating in the ICDAR 2021 conference in terms of success.

Klíčová slova

klasifikácia dokumentov, historické dokumenty, konvolučné neurónové siete, hlboké učenie, attention, masked image modelling, transformer

Keywords

document classification, historical documents, convolutional neural networks, deep learning, attention, masked image modelling, transformer

Citace

PINKEOVÁ, Bettina. *Klasifikace historických dokumentů pomocí hlubokých neuronových sítí.* Brno, 2023. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Kišš

Klasifikace historických dokumentů pomocí hlubokých neuronových sítí

Prohlášení

Prohlašuji, že jsem tuto diplomovú práci vypracovala samostatně pod vedením pana Ing. Martina Kišša. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....
Bettina Pinkeová
16. května 2023

Poděkování

Chcela by som poďakovať pánovi Ing. Martinovi Kiššovi za odbornú pomoc a cenné rady, ktoré mi túto prácu pomohli vypracovať.

Obsah

Kapitola 1

Úvod

V súčasnej dobe najmä kvôli popularite internetu a technológií archivujú digitálne knižnice už mnoho historických dokumentov v digitálnej forme. Dôvodom je aj jednoduchý prístup, ale hlavne možnosť ich zachovania v rozumnej podobe, pretože časom sú dané dokumenty náchylné na postupné znehodnocovanie a toto sa považuje za obrovskú prekážku v ich archivácii. Takýto spôsob uskladňovania dokumentov prináša radu výhod a jednou z nich je prevedenie klasifikácie v oblasti strojového učenia [5].

Klasikifácia historických dokumentov je dôležitou úlohou. Prináša rozumné indexovanie a spravovanie dokumentov v digitálnych knižniciach [5]. K tomu, aby sme o danom dokumente mali čo najviac informácií a mohli ho podrobnejšie skúmať, ho potrebujeme klasifikovať na základe rôznych kritérií, ako napríklad obdobie alebo miesto jeho vzniku. Táto automatizovaná klasifikácia prináša mnoho výhod, ako je zjednodušenie triedenia veľkého množstva dokumentov do stanovených kategórií, čo nám dopomôže k tomu, aby ich bolo možné naďalej rôznymi spôsobmi spracovávať a študovať [16]. Je tiež predpokladom pre rozpoznávanie rukou písaného textu, automatizované indexovanie a dolovanie údajov [16]. Práve kvôli tomu je významným objektom štúdia v oblasti strojového učenia. Vyplýva to aj najmä z konajúcich sa súťaží, čo spôsobilo prínos rôznych metód.

Táto práca sa zameriava na klasifikáciu historických dokumentov pomocou hlbokých neurónových sietí a sústreďuje sa na úlohy: typ skupiny fontu, typ písaného písma, obdobie vzniku, ale predovšetkým na miesto vzniku, nakoľko táto úloha sa radila v rámci usporiadaných súťaží za najmenej úspešnú. V rámci tejto úlohy sú v práci použité konvolučné neurónové siete a vizuálne transformery. Použité architektúry sú opísané v kapitole 2. Táto kapitola sa venuje aj opisu existujúcich metód vychádzajúcich zo súťaží. Ďalšia kapitola 3 predstavuje dostupné dátové sady historických dokumentov pre spomínané úlohy. V kapitole 4 sa nachádza návrh riešení vychádzajúci z analýzy, pričom implementácia jednotlivých návrhov je opísaná v ďalšej kapitole 5. Experimenty s výsledkami následne opisuje kapitola 6.

Kapitola 2

Klasifikácia dokumentov

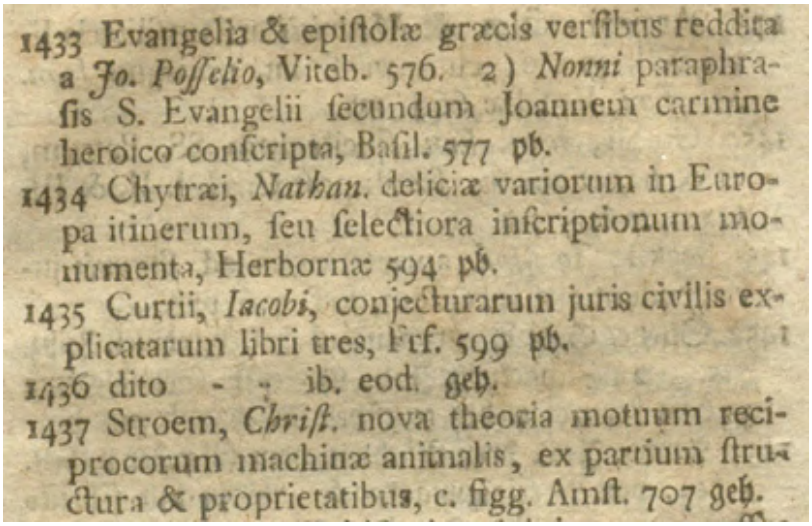
Klasifikačné prístupy na klasifikáciu dokumentov možno zoskupiť do dvoch kategórií, založené na obraze a založené na obsahu, v niektorých prípadoch sa často používa aj kombinácia týchto dvoch prístupov. Ktorý prístup je vhodnejší, závisí hlavne na type dokumentov a klasifikačnej úlohy [15].

Práca sa zaoberá vizuálnou klasifikáciou dokumentov a analyzuje niekoľko úloh. Jednou z nich je určenie typu písma, ktorým je dokument písaný. Rozlišujú sa dva prípady na základe toho, či je dokument písaný písaným alebo tlačeným písmom. Znalosť tejto informácie je užitočná pri výbere adekvátnych modelov pre optické rozpoznávanie znakov alebo ručne písaného písma na získanie textového obsahu dokumentov [24]. Väčšina skupín fontov sa objavila v prvých desaťročiach tlače. Prvá zásadne nová skupina písiem boli bezpätkové písma v 19. storočí. Dovtedy sa nové fonty zriedkavo vyvinuli mimo štylistických hraníc existujúcich skupín písiem. Existujú už, ale tisíce štylisticky podobných fontov, ktoré historici vedia rozdeliť do skupín [21]. Táto úloha môže byť typicky náročnejšia, pretože mnoho dokumentov obsahuje viacero typov fontu v rámci jednej strany a to znamená, že jedna strana môže byť anotovaná viacerými anotáciami.

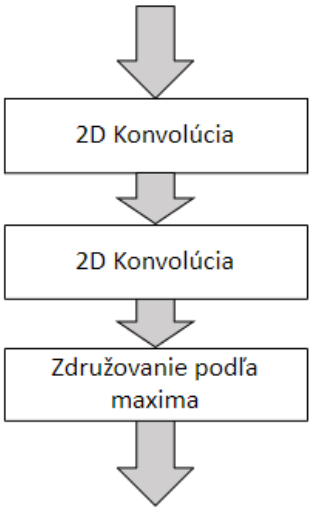
Ďalšou úlohou je datovanie dokumentov do obdobia ich vzniku, čo umožňuje prepojiť dokument s jeho historickým kontextom [4]. Vo väčšine prípadov nie je úplne isté presné obdobie. Toto taktiež môže viesť k výzve, pretože dokumentom sú na základe analýzy priradené intervaly období.

Poslednou úlohou je určenie miesta pôvodu historických dokumentov, ktorá sa ukázala byť najnáročnejšou úlohou, z tohto dôvodu sa na ňu práca najviac sústreďuje. Pri datovaní alebo lokalizácii dokumentov môže ísť aj o dokumenty, ktoré reprezentujú nejaké kresby alebo maľby, nemusí isť len o text.

Na tieto úlohy a klasifikáciu dokumentov podľa obrazu sa vo všeobecnosti používajú konvolučné neurónové siete, ktoré predstavujú najpopulárnejší nástroj v tejto oblasti. Sieť ResNet je stále najpoužívanejšou sieťou v oblasti počítačového videnia, rovnako sa stále používajú aj siete založené na architektúre VGGNet. V súčasnosti sa v rámci počítačového videnia rozšírilo aj použitie vizuálnych transformerov, ktoré vychádzajú z transformerov pre spracovanie prirodzeného jazyka. Rovnako ako pre spracovanie prirodzeného jazyka sa osvedčili aj pre klasifikáciu obrázkov. Táto kapitola stručne opisuje architektúry používané pre klasifikáciu obrázkov a existujúce metódy pre klasifikáciu historických dokumentov.


1433 Evangelia & epiſtolæ græcis verſibus reddita a *Jo. Poſſelio*, Viteb. 576. 2) *Nonni* paraphraſis S. Evangelii ſecundum Joannem carmine heroico conſcripta, Baſil. 577 pb.
1434 Chytræi, *Nathan.* deliciæ variorum in Europa itinerum, ſeu ſelectiora inſcriptionum monumenta, Herbornæ 594 pb.
1435 Curtii, *Iacobi*, conjecturarum juris civilis explicatarum libri tres, Frf. 599 pb.
1436 dito - - ib. eod. geb.
1437 Stroem, *Chriſt.* nova theoria motuum reciprocorum machinæ animalis, ex partium ſtructura & proprietatibus, c. figg. Amſt. 707 geb.

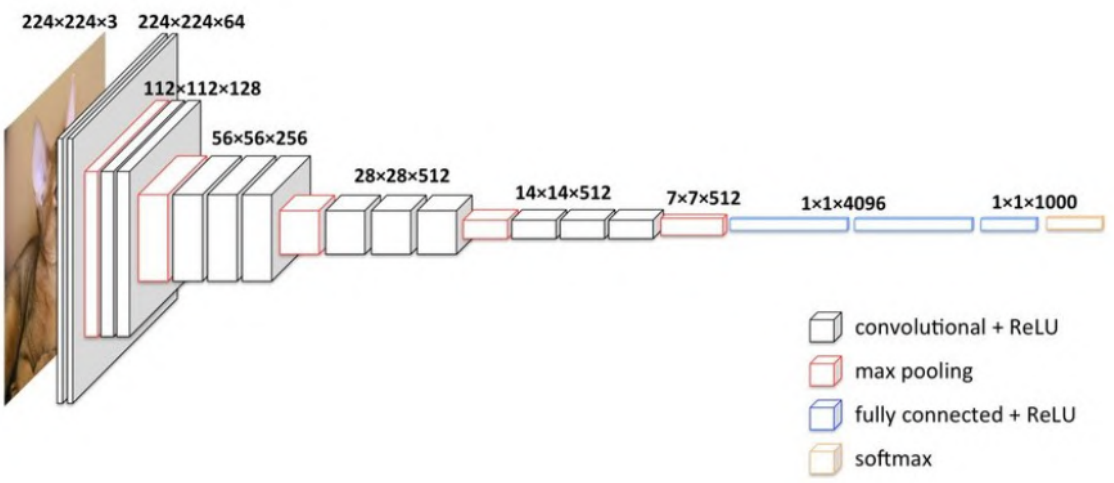
Obrázek 2.1: Príklad strany historického dokumentu, písaného viacerými druhmi fontu. Obrázok obsahuje 4 typy fontu: Fraktur, Italic, Schwabacher a Antiqua. Prevzaté z [22].



Obrázek 2.2: Jeden konvolučný blok architektúry VGGNet.

2.1 VGGNet

Architektúra VGGNet - *Visual Geometry Group Network*, bola prvýkrát predstavená v článku *Very Deep Convolutional Networks for Large-Scale Image Recognition* [25] skupinou výskumníkov z Oxfordu. Cieľom bolo zvýšiť komplexnosť modelu a tým aj jeho výkon. Ide o prelomový model v oblasti počítačového videnia, ktorý je v súčastnosti stále jeden z najpopulárnejších architektúr používajúcich sa v tejto oblasti. Nahradenie veľkých rozmerov konvolučných filtrov za filtre s menšími rozmermi ukázalo výrazné zlepšenie oproti architektúre AlexNet, ktorá mala rozmery filtrov v prvej vrstve veľkosti 11×11 pixelov a v druhej vrstve veľkosti 5×5 pixelov. Architektúra VGGNet je dostupná v dvoch verziách: VGG-16 a VGG-19, pričom čísla 16 a 19 predstavujú počet vrstiev, z ktorých tri vrstvy predstavujú plne prepojené vrstvy [30]. Tieto siete sú založené na klasických konvolučných neuróno-



Obrázek 2.3: Architektúra siete VGG-16[1]. Sivé bloky predstavujú konvolučné vrstvy s následnou aplikáciou funkcie ReLU. Červené bloky predstavujú združovacie vrstvy podľa maxima. Modré bloky znázorňujú plne prepojenú vrstvu a žltý blok je funkcia *softmax*.

vých sieťach. Sú tvorené z blokov, pričom každý blok pozostáva z 2D konvolúcie a z vrstvy združovania podľa maxima [20]. Konvolučný blok architektúry VGGNet je znázornený na obrázku 2.2. Jednou zo zásadných nevýhod tejto architektúry je, že ide o obrovskú sieť, čo znamená, že natrénovanie jej parametrov zaberie väčšie množstvo času. V nasledujúcej časti sa nachádza opis siete VGG-16, ktorá je najpoužívanejšou sieťou architektúry VGGNet.

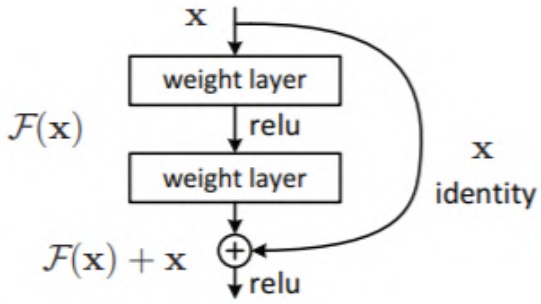
VGG-16

Vstupom do konvolučnej neurónovej siete architektúry VGG-16 je obrázok s fixnou veľkosťou 224×224 pixelov v RGB prevedení. Následne obrázok prechádza cez hromadu konvolučných vrstiev. Pri konvolúcii v týchto vrstvách sú použité konvolučné filtre s veľmi malým vnímavým poľom, ako už bolo spomenuté v prvej časti tejto kapitoly. Konkrétne ide o filtre veľkosti 3×3 pixelov. Táto veľkosť predstavuje najmenšiu možnú veľkosť na zachytenie okolia [20]. V jednej z konfigurácií dokonca architektúra VGG-16 používa konvolučné filtre aj veľkosti 1×1, čo možno pokladať za jednoduchú lineárnu transformáciu vstupných kanálov s následnou aplikáciou nelinearity. Hodnota posunu konvolúcie je pevne stanovená na 1 pixel. Priestorová výplň vstupu konvolučnej vrstvy predstavuje 1 pixel, čo znamená, že priestorové rozlíšenie sa po konvolúcii zachováva [30]. Po každej konvolučnej vrstve je aplikovaná funkcia ReLU a má nasledovný tvar, kde x predstavuje hodnotu daného pixelu [20]:

$$ReLU(x) = max(0, x)$$

Priestorové združovanie sa uskutočnuje prostredníctvom piatich združovacích vrstiev, ktoré aplikujú zdužovanie podľa maxima - *max pooling*. Združovanie podľa maxima sa vykonáva prostredníctvom okna veľkosti 2×2 pixelov, pri veľkosti posunu nastavenej na 2 pixely [26]. Architektúra teda obsahuje hromadu konvolučných a združovacích vrstiev, po ktorých nasledujú tri plne pripojené vrstvy: prvé dve plne prepojené vrstvy majú každá

[1] https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/



Obrázek 2.4: Stavebný blok reziduálneho učenia. Prevzaté z [10].

4096 kanálov a tretia vykonáva klasifikáciu do 1000 tried a teda obsahuje 1000 kanálov, čo je jeden kanál pre každú triedu. Posledná vrstva je *softmax*, ktorej výstupom sú pravdepodobnosti jednotlivých tried. Konfigurácia plne prepojených vrstiev je podobná vo všetkých sieťach architektúry VGGNet. Architektúra VGG-16 je bližšie znázornená na obrázku 2.3.

2.2 ResNet

Pridávanie veľkého množstva vrstiev do neurónových sietí za cieľom riešenia komplexnejších problémov prináša problémy pri trénovaní. Veľmi hlboké siete sú náročné na trénovanie a týmto klesá aj ich úspešnosť. Opisovaná architektúra ResNet - *Residual Neural Network*, pomáha vyriešiť tento problém a je jednou z najpoužívanejších architektúr v oblasti hlbokého učenia.

Kaiming He, Xiangyu Zhang, Shaoqing Ren a Jian Sun predstavili ResNet v roku 2015 v článku *Deep Residual Learning for Image Recognition* [10]. Navrhli tzv. *deep residual learning framework*, ktorý rieši problém miznúceho gradientu pri trénovaní hlbokých sietí a tým predchádza degradácii presnosti modelu. Táto architektúra je zložená z tzv. reziduálnych blokov [27], ktoré sú v nasledujúcej časti kapitoly opísané.

$H(x)$ predstavuje zobrazenie obsahujúce niekoľko vrstiev siete naskladaných na seba, kde x označuje vstup do prvej z vrstiev. Keďže sa predpokladá, že viacero nelineárnych vrstiev môže asymptoticky aproximovať komplikované funkcie, je možné predpokladať, že aproximujú aj tzv. reziduálne funkcie $H(x) - x$, za predpokladu, že vstup aj výstup majú rovnaké rozmery [10]. Takže namiesto toho, aby sa očakávalo, že vrstvy budú aproximovať $H(x)$, sa tieto vrstvy explicitne nechajú aproximovať reziduálnu funkciu $F(x) = H(x) - x$ a z pôvodnej funkcie sa teda stáva funkcia $F(x) + x$ [10]. Táto preformulácia je motivovaná spomínaným problémom degradácie. Ak by pridávané vrstvy do siete mohli byť konštruované ako mapovania identity, chybovosť hlbšieho modelu s pridanými vrstvami by nemala byť väčšia, ako má jeho plytkejší predchodca. Problém degradácie však nasvädčuje tomu, že nastáva problém pri aproximácii viacerými nelineárnymi vrstvami, funkcie predstavujúce mapovanie identity [28]. Pri reziduálnom učení, ak sú mapovania identity optimálne, sa môžu prestaviť váhy viacerých nelineárnych vrstiev k hodnotám približujúcim sa k nule, za účelom priblíženia sa k identickému mapovaniu. V reálnych podmienkach je vysoko nepravdepodobné, že by boli mapovania identity optimálne, ale táto preformulácia môže pomôcť pri riešení spomínaného problému [28]. Reziduálne učenie je aplikované vždy na niekoľko vrstiev naskladaných na seba, čo predstavuje stavebný blok, matematicky definovaný na-

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				

Obrázek 2.5: Porovnanie všetkých architektúr ResNet podľa počtu vrstiev. Prevzaté z [10].

sledovne, kde x a y sú vstupné a výstupné vektory uvažovaných vrstiev a funkcia $\mathcal{F}(x, W_i)$, predstavuje žiaduce reziduálne mapovanie [10]:

$$y = \mathcal{F}(x, W_i) + x$$

Na obrázku 2.4 blok predstavuje dve vrstvy a funkcia $\mathcal{F}(x, W_i)$ sa matematicky dá definovať ako: $\mathcal{F} = W_2\sigma(W_1x)$, kde σ predstavuje funkciu ReLU. Opeácia $\mathcal{F} + x$ je realizovaná prostredníctvom spojenia predstavujúce skratku a elementárneho sčítavania [10]. Po sčítaní sa znova aplikuje nelinearita. Spojenia prostredníctvom skratiek nie sú náročné na parametre ani na výpočtovú zložitosť. V prípade prvej rovnice musia byť rozmery $\mathcal{F}$ a x rovnaké, ak tomu tak nie je platí [10]:

$$y = \mathcal{F}(x, W_i) + W_sx$$

Tvar funkcie F môže byť ľubovoľný a môže reprezentovať aj viacero konvolučných vrstiev. Sieť architektúry ResNet, ktorá využíva reziduálne učenie, môže pozostávať z ľubovoľného počtu vrstiev. V článku [10] navrhli sieť s 34 vrstvami. Sieť je inšpirovaná architektúrou VGG-19, do ktorej sú následne pridané prepojenia reprezentujúce skratky. Využívajú sa však aj architektúry ResNet-18, ResNet-50, Resnet-101 a ďalšie verzie tejto architektúry s rôznym počtom vrstiev [27].

ResNet-50

Architektúra ResNet-50 sa zvyčajne používa najčastejšie pre úlohy klasifikácie obrázkov. Pozostáva zo štyroch úrovní, pričom každá úroveň obsahuje rôzny počet zvyškových blokov v tomto poradí: 3, 4, 6 a 3, z ktorých každý obsahuje tri vrstvy. Sieť môže prijať vstupný obraz s rozmermi, ktoré sú násobkami čísla 32 a počtom kanálov rovným trom. V prípade ak má sieť na vstupe obrázok veľkosti $H \times W \times 3$, kde H a W predstavujú rozmery vstupného obrázka, sa aplikuje počiatočná konvolúcia a združovanie podľa maxima za použitia filtrov veľkosti 7×7 a 3×3 pixelov, pričom hodnota posunu pre obe operácie je rovná dvom pixelom [19]. Následne sa uplatňujú jednotlivé fázy siete. Keďže operácia konvolúcie vo zvyškovom bloku sa vykonáva s hodnotou posunu 2 pixely, veľkosť vstupu sa zníži na polovicu, pokiaľ

ide o výšku a šírku, ale počet kanálov sa zdvojnásobí po prechode jednej fázy do ďalšej [19]. Ako už bolo spomenuté, v každom zvyškovom bloku sú na seba naukladané tri vrstvy. Ide o vrstvy s operáciou konvolúcie s filtrami veľkosti 1×1, 3×3 a 1×1 pixelov.

Na konci siete sa na výstup zo všetkých fáz aplikuje združovanie podľa priemeru - *global average pooling* a nasleduje plne prepojená vrstva s funkciou *softmax*, ktorej výstupom sú pravdepodobnosti pre 1000 tried [19]. Na obrázku 2.5 sú podrobnejšie znázornené jednotlivé fázy pre každú sieť architektúry ResNet.

2.3 Transformer

Architektúry založené na *self-attention* mechanizme, najmä transformery, sa stali prvou voľbou v oblasti spracovania prirodzeného jazyka. Dominantným prístupom je predtrénovanie na veľkom textovom korpuse a následné doladenie na menšiu dátovú sadu, ktorá je špecifická pre danú úlohu [6]. Na základe veľkého úspechu transformerov v oblasti spracovania prirodzeného jazyka Dosovitskiy et al. v článku *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale* [8] predstavili tzv. vizuálny transformer - *Vision transformer*.

Vizuálny transformer vychádza z toho pôvodného transformeru opísaného v článku *Attention is All You Need* [29]. Jeho architektúra je znázornená na obrázku 2.6. Štandardný transformer prijíma na vstupe sekvenciu tokenových embedingov. K tomu, aby sa dokázali spracovať dvojrozmerné obrázky, sa upravia dimenzie obrázka x z pôvodného tvaru $H \times W \times C$ na sekvenciu vektorizovaných výrezov x_P z pôvodného obrázka, veľkosti $N \times (P^2 * C)$. (H, W) je v tomto prípade rozlíšenie pôvodného obrázka, C predstavuje počet kanálov a (P, P) vyjadruje rozlíšenie výrezov [8]. Počet výrezov je možné jednoducho vyjadriť nasledujúcim vzorcom, kde N tiež predstavuje aj dĺžku vstupnej sekvencie pre transformer [29].

$$N = (H * W)/P^2$$

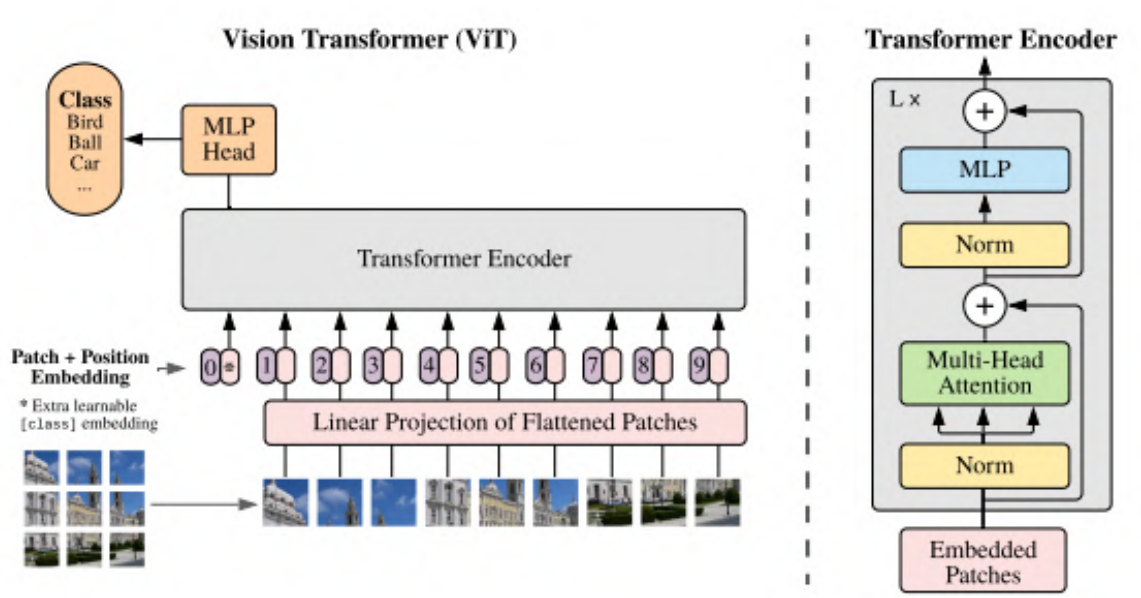
Transformer používa rovnakú veľkosť vektora D naprieč všetkým svojim vrstvám, takže sa výrezy vektorizujú a následne sa vykoná mapovanie do D dimenzií s trénovateľnou lineárnou projekciou. Výstupom tejto projekcie sú embedingy výrezov. K týmto embedingom sú pridané tzv. jednorozmerné pozičné embedingy (E_{pos}), ktoré pomáhajú zachovávať informáciu o pozícii výrezu a ďalší trénovateľný embeding ($z_0^0 = x_{class}$), ktorý sa umiestni na prvú pozíciu danej vstupnej sekvencie. Tento embeding nie je asociovaný so žiadným výrezom, výstup pre tento embeding sa použije ku klasifikácii. Výsledná sekvencia embedingov slúži ako vstup do enkódera transfomeru. Operácie sú matematicky vyjadrené nasledovne [8]:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; ...; x_p^N E] + E_{pos}$$

$$E \in \mathbb{R}^{(P^2 * C) \times D}$$

$$E_{pos} \in \mathbb{R}^{(N+1) \times D}$$

Stav na výstupe enkódera transformeru (Z_L^0) slúži ako obrazová reprezentácia y [8]. Či už v prípade predtrénovania alebo doladovania je implementovaná klasifikačná vrstva predstavujúca viacvrstvový perceptrón - *Multilayer perceptron* (MLP). V čase predtrénovania má tento viacvrstvový perceptrón jednu skrytú vrstvu. V prípade doladovania, ide o jedinú lineárnu vrstvu [29].



Obrázek 2.6: Architektúra vizuálneho transformeru spoločne s enkóderom (vpravo). Obrázok sa rozdelí na výrezy fixnej veľkosti, po lineárnej projekcii sa k embedingom výrezov pridá pozičný embeding. Vznikne sekvencia, ktorá je vstupom do enkódera. Na vykonanie klasifikácie sa používa štandardný prístup pridania ďalšieho trénovateľného parametra tzv. klasifikačného tokenu do sekvencie [8]. Prevzaté z [8].

$$z'_\ell = MSA(LN(z_{\ell-1})) + z_{\ell-1}$$

$$z_\ell = MLP(LN(z'_\ell)) + z'_\ell$$

$$y = LN(z_0^L)$$

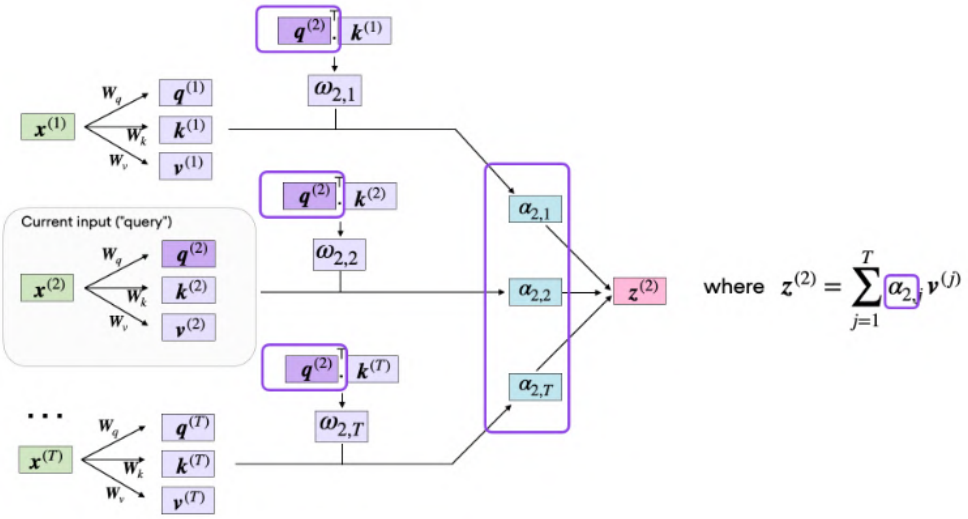
$$\ell = 1...L$$

Enkóder predstavuje súbor L indetických vrstiev. Každá vrstva je zložená z dvoch podvrstiev: *multihead self-attention* modulu (MSA) a MLP blokov [29]. Normalizačná vrstva (LN) sa nachádza pred každým blokom, pričom reziduálne prepojenia sú implementované po každom bloku. Viacvrstvový perceptrén obsahuje dve vrstvy aplikujúce nelinearitu prostredníctvom aktivačnej funkcie GELU - *Gaussian Error Linear Units*, ktorá má nasledovný tvar, kde x predstavuje danú hodnotu, na ktorú sa nelinearita aplikuje [8]:

$$GELU(x) = 0.5 * x * (1 + tanh(\tfrac{2}{\pi}) * (x + 0.044715 * x^3)$$

Self-Attention

Koncept *self-attention* v hlbokom učení má svoje korene v snahe zlepšiť rekurentné neurónové siete za účelom spracovania dlhších sekvencií. V súčastnosti je tento mechanizmus obľúbeným stavebým blokom architektúr neurónových sietí. Umožňuje modelu merať dôležitosť jednotlivých prvkov vo vstupnej postupnosti a na základe tejto dôležitosti, dymanicky upravovať mieru ich vplyvu na výstup [31].



Obrázek 2.7: Ľavá strana: Získavanie sekvencie dotazov (q), kľúčov (k) a hodnôt (v) pre každý prvok vstupnej postupnosti x prostredníctvom matíc váh W. Stred: Výpočet nenormalizovaných *attention* váh ω, prostredníctvom q a k. Pravá strana: Normalizácia váh použitím funkcie *softmax* a následný výpočet kontextového vektoru z. Obrázok znázorňuje príklad výpočtu *attention* váh a kontextového vektora pre $i = 2$ [31]. Prevzaté z [31].

Self-attention využíva tri typy matice váh označované ako W_q, W_k a W_v, ktoré predstavujú parametre modelu a upravujú sa počas trénovania. Tieto matice slúžia na premietnutie vstupov do nasledovných komponentov: dotaz, kľúč a hodnota [31]. Príslušné sekvencie dotazov, kľúčov a hodnôt sa získajú násobením medzi maticami váh W a vstupmi x nasledovne:

$$q^{(i)} = W_q x^{(i)}; i \in [1, T]$$
$$k^{(i)} = W_k x^{(i)}; i \in [1, T]$$
$$v^{(i)} = W_v x^{(i)}; i \in [1, T]$$

Index i označuje pozíciu prvku vo vstupnej sekvencii, ktorá má dĺžku T, q predstavuje sekvenciu dotazov, k predstavuje sekvenciu kľúčov a v sekvenciu hodnôt. V tomto prípade $q^{(i)}$ a $k^{(i)}$ sú vektory veľkosti d_k, nakoľko v nasledujúcom kroku sa bude vykonávať skalárny súčin medzi vektormi dotazu a vektormi kľúča, musí platiť $d_k = d_q$ [31]. Rozmer matíc W_k a W_q je $d_k \times d$ a $d_v \times d$ predstavuje rozmer matice W_v, kde d predstavuje veľkosť každého vektora [29].

Ďalším krokom je výpočet nenormalizovaných *attention* váh označených ω. Na obrázku 2.7 je znázornené, že *attention* váhy sa počítajú ako skalárny súčin medzi sekvenciami dotazu a kľúča [31].

$$\omega_{ij} = q^{(i)^T} k^j$$

Následne sa váhy ω normalizujú, aby sa získali normalizované *attention* váhy α prostredníctvom funkcie *softmax*. Navyše ešte pred použitím funkcie *softmax*, sú váhy škálované násobkom $1/\sqrt{d_k}$ nasledovne [31]:

$$\alpha_{ij} = softmax(\frac{\omega_{ij}}{\sqrt{d_k}})$$

Škálovanie podľa $1/\sqrt{d_k}$ zabezpečuje že Euklidovská vzialenosť vektorov váh bude približne v rovnakom rozsahu. To pomáha zabrániť tomu, aby sa *attention* váhy stali príliš malými alebo naopak, príliš veľkými, čo by mohlo viesť k numerickej nestabilite alebo ovplyvniť schopnosť modelu konvergovať počas trénovania [29].

Posledný krok zahŕňa výpočet kontextového vektora $z^{(i)}$. Tento vektor predstavuje váhovaný pôvodný vstup $x^{(i)}$, prostredníctvom *attention* váh, vrátane všetkých ostatných vstupných prvkov[31].

$$z^{(i)} = \sum_{j=1}^{T} a_{ij} v^{(j)}$$

MultiHead Self-Attention

Self-attention mechanizmus je integrovaný do transformera vo forme tzv. *multihead self-attention* [8]. V rámci tohto mechanizmu, ako už bolo spomínané v predchádzajúcej časti kapitoly, sa vstupná postupnosť transformovala prostredníctvom troch matíc, reprezentujúcich dopyt, kľúč a hodnotu. Tieto tri matice možno považovať za jednu hlavu v kontexte *multihead self-attention*, teda tzv. *self-attention* s viacerými „hlavami“ [8]. Tento typ *self-attention* mechanizmu sa teda skladá z viacerých takýchto hláv, z ktorých každá pozostáva z matíc dotazov, kľúčov a hodnôt. Tento koncept je podobný použitiu viacerých konvolučných filtrov v konvolučných neurónových sieťach [31]. Vstup sa rovnomerne rozdelí medzi tieto hlavy, výpočet sa teda vykonáva paralelne h-krát, pričom h predstavuje počet hláv. Dané výstupy sa následne zkonkatenujú a vykoná sa projekcia [29].

$$SA(q,k,v) = softmax(\frac{q^T k}{\sqrt{d_k}})v$$

$$MultiHead(q,k,v) = Concat(SA_1, SA_2, ...SA_h)w^o$$

SA v tomto prípade predstavuje mechanizmus *self-attention* opísaný v predlošlej kapitole a $MultiHead$ opisuje konkatenáciu výstupov funkcie SA, kde h je rovný počtu hláv resp. opakovaní. Výstupom je ďalšia lineárna transformácia prostredníctvom trénovatelných parametrov w^o [31].

2.4 BEiT

Transformer dosahuje znamenitý výkon v oblasti počítačového videnia. Empirické štúdie však ukazujú, že vizuálne trasformery vyžadujú viac tréningových dát ako konvolučné neurónové siete. Na vyriešenie tohto problému je výhodné aplikovať semi-supervizované predtrénovanie [29].

V súčasnosti dosiahol model BERT - *Bidirectional Encoder Representations from Transformers*, veľký úspech v oblasti spracovania prirodzeného jayzka. Bol prvýkrát predstavený v článku *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* [7], ide o úlohu modelovania maskovaného textu, v rámci ktorého sa najprv náhodne zamaskuje určitú časť tokenov v texte a následne sa maskované tokeny obnovia na základe výsledkov enkódera transformeru, ktorý mal na vstupne neúplný text [2]. Vychádzajúc z tohto prístupu, Hagbo Bao et al. predstavili model BEiT - *Bidirectional Encoder representation from Image Transformers*. Inšpirovaní modelom BERT navrhli predtrénovaciu úlohu modelovanie maskovaných obrázkov [2]. Hlavný myšlienka tejto predtrénovacej úlohy je, že počas trénovania sa najprv náhodne zamaskuje určitá časť výrezov pôvodného obrázka,

poškodený vstup je použitý, ako vstup do transformera a model sa naučí obnoviť maskované časti pôvodného obrázka [2].

Ako základná sieť je použitý vizuálny transformer opisaný v kapitole 2.3. Vstupné obrázky teda budú spracované rovnakým spôsobom, ako v prípade vizuálnych transformerov opísaných v predošlej kapitole. Pri predtrénovaní na túto úlohu sa využívajú teda dva pohľady na obrázky: výrezy obrázkov (viď kapitolu 2.3) a vizuálne tokeny.

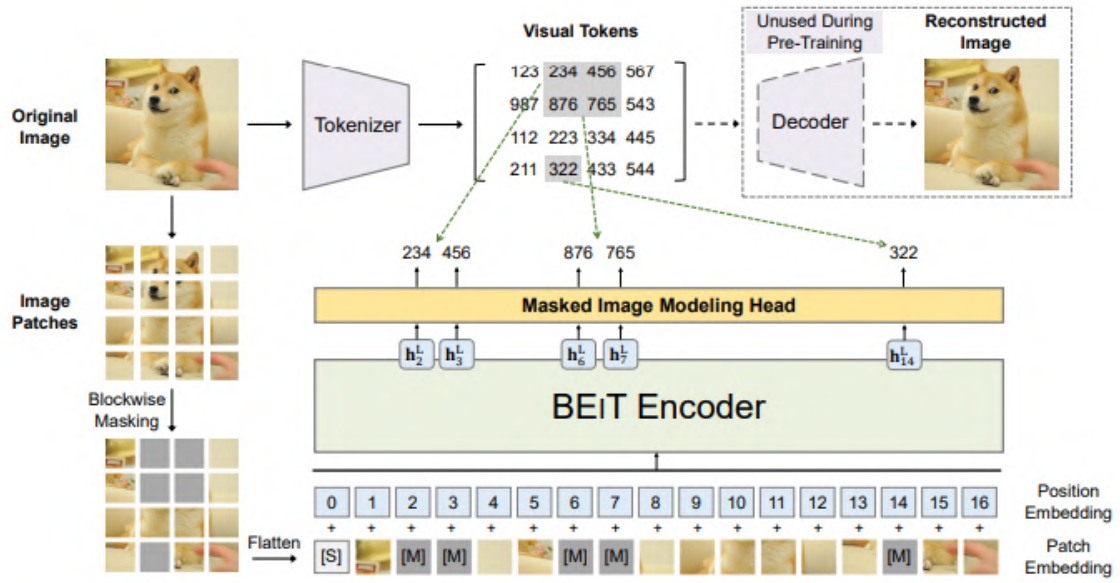
V rámci prvého „pohľadu“, sa obrázok najprv rozdelí na sekvenciu výrezov veľkosti 16×16 pixelov. Vychádzajúc z predošlej kapitoly, obrázok $x \in \mathbb{R}^{H \times W \times C}$ sa teda rozdelí na $N = HW/P^2$ výrezov $x^p \in \mathbb{R}^{N \times P^2 * C}$, kde C predstavuje počet kanálov, (H, W) je rozlíšenie pôvodného obrázka a (P, P) je rolíšenie výrezov. Vstupom transformera je teda sekvencia výrezov z pôvodneho obrázka $\{x_i^p\}_{i=1}^N$ [31]. Výrezy sú lineárne projektované na získanie embedingov Ex_i^p, kde $E \in \mathbb{R}^{(P^2 * C) \times D}$. Okrem toho, sa pridáva špeciálny token $[S]$ na prvú pozíciu danej sekvencie, ako aj štandardný jednodimenzionálny pozičný embeding $E_{pos} \in \mathbb{R}^{(N \times D)}$. Vstupné vektory $H_0 = [e_{[S]}, Ex_i^p, ..., Ex_N^p] + E_{pos}$ predstavujú vstup do enkóderu transformera. Vychádza sa z tzv. embedingov slov v oblasti spracovania prirodzeného jazyka [8]. Ako už bolo spomenuté, enkóder je zložený z L vrstiev blokov transformera $H^l = Transformer(H^{l-1})$, kde $L = 1, ..., L$. Výstupné vektory poslednej vrstvy $H^L = [h_{[S]}^L, h_1^L, ..., h_N^L]$ sa používajú ako zakodované reprezentácie výrezov pôvodného obrázka, kde h_i^L je vektor i-teho výrezu [2].

K tomu, aby transformer vedel predikovať maskovaný vstup, je potrebný druhý „pohľad“ na pôvodný vstupný obrázok. Je teda potrebné vygenerovať „anotácie“. Postup je nasledovný: obrázok sa tokenizuje na tzv. diskrétne vizuálne tokeny. Ide teda o tokenizáciu obrázka $x \in \mathbb{R}^{H \times W \times C}$ na $z = [z_1, z_2, ..., z_N] \in \nu^{h \times w}$, pričom slovník $\nu = 1, ..., |\nu|$ pozostáva z diskrétnych indexov tokenov. Na tokenizovanie je použitý tokenizér obrázkov učený diskrétnym variačným autoenkóderom (dVAE). Priebeh učenia tokenizéra zaisťujú dva moduly, tokenizér a dekóder [18]. Tokenizér $q_\phi(z|x)$ mapuje jednotlivé pixely obrázka x na diskrétnych tokeny z podľa nejakého číselníka, teda tzv. slovníka. Dekóder $p_\psi(x|z)$ sa naučí rekonštruovať vstupný obrázok x na základe vizuálnych tokenov z. Rekonštrukcia sa môže formálne zapísať ako [18]:

$$\mathbb{E}_{z q_{\phi(z|x)}}[log p_\psi(x|z)]$$

Modelovanie maskovaných obrázkov

Metóda predtrénovania, ktorá bola navrhnutá sa volá modelovanie maskovaných obrázkov - *Masked Image Modelling* (MIM). Na obrázku 2.8 je znázornené, že po rozdelení vstupného obrázka x na N výrezov $\{x_i^p\}_{i=1}^N$ a po tokenizácii obrázka na N vizuálnych tokenov $\{z_i\}_{i=1}^N$, je zamaskovaných okolo 40% vstupných výrezov. Tieto pozície maskovaných výrezov sú označené ako $\mathcal{M} \in \{1, ..., N\}^{0.4N}$. Zamaskované výrezy sú nahradené trénovateným enbedingom $e_{[M]} \in \mathbb{R}^D$. Poškodené výrezy $x^{\mathcal{M}} = \{x_i^p : i \notin \mathcal{M}\}_{i=1}^N \bigcup \{e_{[M]} : i \in \mathcal{M}\}_{i=1}^N$ slúžia ako vstup do L vrstvového transformera. Výsledné skryté vektory $\{h_i^L\}_{i=1}^N$ sa považujú za zakódované reprezentácie vstupných výrezov [18]. Pre každú maskovanú pozíciu $\{h_i^L : i \in L\}_{i=1}^N$ sa vykoná klasifikácia prostredníctvom funkcie *softmax* na predikovanie zodpovedajúcich vizuálnych tokenov $p_{MIM}(z^{'}|x^{\mathcal{M}}) = softmax_{z^{'}}(W_c h_i^L + b_c)$, kde $x^{\mathcal{M}}$ je poškodený obrázok, $W_c \in \mathbb{R}^{|\upsilon| \times D}$ a $b_c \in \mathbb{R}^{|\upsilon|}$. V rámci predtrénovania je cieľom maximalizovať *log-likelihood* správnych vizuálnych tokenov vzhľadom na poškodený obrázok nasledovne [2], kde $\mathcal{D}$ je trénovací korpus, $\mathcal{M}$ predstavuje náhodne maskované pozície a $x^{\mathcal{M}}$ je poškodený obrázok , ktorý je maskovaný podľa $\mathcal{M}$ [2]:



Obrázek 2.8: Spôsob predtrénovania modelu BEiT. Počas predtrénovania sa každý obrázok rozdelí na výrezy a vykoná sa transformácia pôvodného obrázka na diskrétne vizuálne tokeny. Dôjde k náhodnému maskovaniu určitého podielu výrezov (sivé políčka na obrázku) a nahradeniu ich špeciálnym embedingom [M]. Potom výrezy vstupujú do transformera spolu s pozičným embedingom a tzv. klasifikačným tokenom. Predtrénovacia úloha sa zameriava na predikovanie vizuálnych tokenov pôvodného obrázka. Tieto predikcie sa porovnávajú s diskrétnymi vizuálnymi toknemi vzniknuté pomocou tokenizéra [2]. Prevzaté z [2].

$$\max \sum_{x \in \mathcal{D}} \mathbb{E}_{\mathcal{M}} \left[\sum_{i \in \mathcal{M}} log p_{MIM}(z_i | x^{\mathcal{M}}) \right]$$

Narozdiel od modelu BERT pre spracovanie prirodzeného jazyka, kde výber maskovaných tokenov prebieha náhodne, v tomto prípade prebieha výber maskovaných výrezov prostredníctvom metódy blokového maskovania. Takýto spôsob maskovania sa ukázal v tomto prípade ako efektívnejší. Hlavná myšlienka spočíva v tom, že sa iteratívne vyberá blok výrezov, ktoré budú predstavovať maskované výrezy. Najprv je nastavený minimálny počet výrezov, následne sa náhodne vyberie pomer šírky k výške maskovaného bloku. Tieto kroky sa opakujú, kým nie je k dispozícii dostatočný počet maskovaných výrezov, v tomto prípade $0.4N$, pričom N predstavuje celkový počet výrezov a 0.4 je maskovací pomer [18]. Algoritmus je bližšie vyjadrený na 2.9.

Klasifikácia

Po predtrénovaní, sa v rámci dolaďovania pre klasifikačné úlohy pripojí klasifikačná vrstva, ktorá predstavuje jednoduchý lineárny klasifikátor. Konkrétne sa používa združovanie podľa priemeru na agregáciu výstupu a agregovaný výstup je vstupom do lineárnej vrstvy a následne do funkcie *softmax*. Pravdepodonosti jednotlivých tried je možné vyjadriť nasledovne, kde h_L^i je výstupný vektor i-teho výrezu, $W_C \in \mathcal{R}^{D \times C}$ je matica parametrov a C je počet tried [2]:

Algorithm 1 Blockwise Masking

Input: $N(= h \times w)$ image patches
Output: Masked positions $\mathcal{M}$
$\mathcal{M} \leftarrow \{\}$
repeat
  $s \leftarrow \text{Rand}(16, 0.4N - |\mathcal{M}|)$ ▷ *Block size*
  $r \leftarrow \text{Rand}(0.3, \frac{1}{0.3})$ ▷ *Aspect ratio of block*
  $a \leftarrow \sqrt{s \cdot r}; b \leftarrow \sqrt{s/r}$
  $t \leftarrow \text{Rand}(0, h - a)\,; l \leftarrow \text{Rand}(0, w - b)$
  $\mathcal{M} \leftarrow \mathcal{M} \bigcup \{(i, j) : i \in [t, t + a), j \in [l, l + b)\}$
until $|\mathcal{M}| > 0.4N$ ▷ *Masking ratio is 40%*
return $\mathcal{M}$

Obrázek 2.9: Ukážka algoritmu blokového maskovania. Prevzaté z [2].

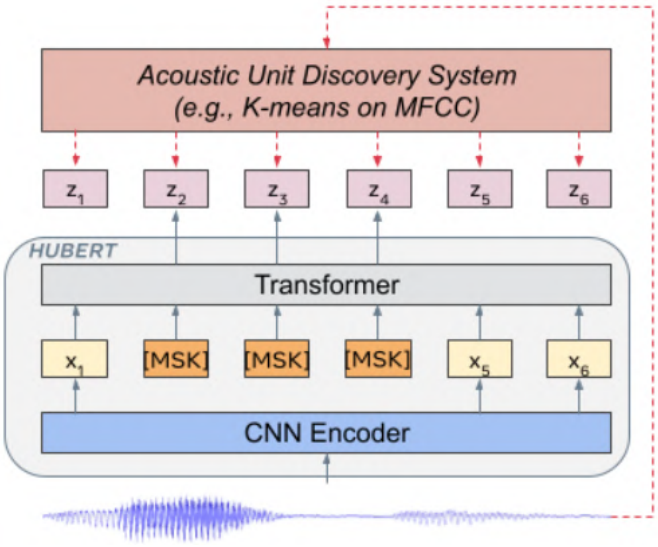
$$\text{softmax}(\text{avg}\ (\{\text{h}_L^i\}_{i=1}^N W_c))$$

HuBERT Na podobnom princípe vychádzajúcom z modelu BERT pre spracovanie prirodzeného jazyka, je v článku *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units* [11], navrhnutý prístup semi-supervizovaného učenia prostredníctvom modelovania maskovanej reči. Prístup bol navrhnutý z dôvodu, že rečové signály sa líšia od textu a obrázkov, pretože ide o sekvencie so spojitou hodnotou. Nejde síce o obrázkový prístup, ale v tejto práci sa relizuje určitá časť tohto návrhu popísaného v nasledujúcej časti.

Trénovanie spočíva taktiež v dvoch častiach. V generovaní diskrétnych tokenov reprezentujúcich reč, ktoré v článku nazývajú skryté jednotky. Druhá časť spočíva v snahe predikovať tieto skryté jednotky zo vstupu predstavujúceho maskovanú reč [11]. Na tokenizáciu sekvencie reči navrhli použiť algoritmus k-means, ktorý aplikujú na extrahované MFCC príznaky. Nech X predstavuje sekvenciu extrahovaných MFCC príznakov, $X = [x_1, x_2, ..., x_T]$, kde T predstavuje počet rámcov. Objavené skryté jednotky sú označené ako $h(X) = Z = [z_1, z_2, ..., z_T]$, kde $z_t \in [C]$ je kategorická premenná prislúchajúca triede C a h predstavuje model na báze zhlukovania, *k-means*. Fáza trénovania transformeru na maskovaných rámcoch vstupnej sekvencie následne prebieha podobne, ako v ostatných prístupoch [11] (viď obrázok 2.10).

2.5 Klasifikácia historických dokumentov

Nasledujúca kapitola popisuje existujúce prístupy klasifikácie historických dokumentov. Popísané prístupy vychádzajú z metód, ktoré boli zrealizované na podnet súťaží [4], [24] a [3]. Ide o konferencie, ktoré realizovali úlohy: datovanie a lokalizácia dokumentov a klasifikácia typu písma. Tieto prístupy boli následne evaluované na príslušných testovacích dátových sadách organizátormi súťaže. Kapitola sa venuje hlavne najúspešnejším systémom. Väčšinu existujúcich metód klasifikácie historických dokumentov možno rozdeliť do dvoch kategórií. Metódy používajúce architektúru ResNet a metódy založené na architektúre VGGNet. Tieto architektúry sú opísané v kapitole 2.1 a 2.2.

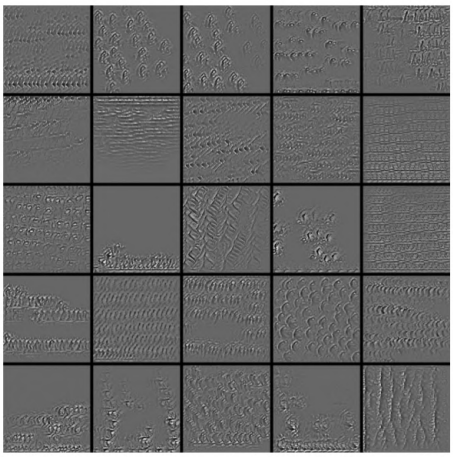


Obrázek 2.10: Prístup HuBERT predikuje skryté jednotky, teda priradenia klastrov vygenerovaných maskovaným rámcom jednou alebo viacerými iteráciami zhlukovania modelom *k-means* [11]. Prevzaté z [11].

DeepScript Najúspešnejšou metódou vypracovanou na podnet súťaže na koferencii ICFHR 2016 - *International Conference on Frontiers in Handwriting Recognition* [4], bola metóda postavená práve na spomínanej architektúre VGGNet. Usporiadaná súťaž bola zameraná na úlohu klasifikácie historických rukopisov datovaných do obdobia stredoveku - *Competition on the Classification of Medieval Handwritings in Latin Script*. Išlo konkrétne o určenie typu písaného písma týchto dokumentov. Víťazný systém DeepScript, pochádza z univerzity v Antverpoch. Systém je založený na variante architektúry VGG-16, pracujúcej s čiernobielymi obrázkami, teda s obrázkami rozmerov $H \times W \times 1$, kde H, W sú rozmery a 1 predstavuje počet kanálov. Použitý model danej architektúry má štandardnú podobu súboru konvolučných vrstiev so stúpajúcim počtom filtrov v každom bloku vrstiev. Tento súbor sa napája do dvoch plne prepojených vrstiev s dimenzionalitou veľkosti 1048 pred funkciou *softmax* [4]. Pre regularizáciu pri trénovaní siete bola použitá tzv. technika *dropout*, čo je technika učenia, ktorá spočíva v tom, že behom učenia sa niektoré neuróny vyradia z činnosti, tým pádom sa odstránia všetky prepojenia z nich alebo do nich vedúce. Keďže dodaná dátová sada bola relatívne malá, bola augmentačná stratégia nasledovná: počas trénovania, po každej epoche bol každý obrázok zmenšený o faktor veľkosti dva a bolo náhodne vybraných 100 výrezov veľkosti 300×300 pixelov [13]. Na tieto výrezy bola aplikovaná dátová augmentácia, ktorá zahŕňala napr. zmenu uhlu alebo posun. Týmto sa po každej epoche vytvorili náhodné trénovacie vzory veľkosti 150×150 pixelov. Anotácia každého výrezu predstavuje triedu strany dokumentu, z ktorej bol výrez extrahovaný [13].

Výsledná predikcia siete na úrovni strany dokumentu prebieha nasledovne: z každého obrázka bolo náhodne vybraných 30 výrezov veľkosti 150×150 pixelov. Výstupy siete pre tieto výrezy boli následne spriemerované pre jednotnú predikciu [4].

Mimoriadne zaujímavá bola schopnosť vizualizovať znalosti odvodené natrénovanou neurónovou sieťou, teda vizualizácia toho, na aký druh informácií sa stala sieť citlivou [4]. Ukážka tejto vizualizácie je viditeľná na obrázku 2.11.



Obrázek 2.11: Ukážka vizualizácií filtra, ktorý zvýrazňuje dôležité informácie vo výrezoch. Vizualizácia bola dosiahnutá pomocou princípu gradientového vzostupu, aby sa maximálne aktivovali jednotlivé neuróny v poslednej konvolučnej vrstve. Vizualizácia znázorňuje, podľa ktorých informácií sa sieť najviac rozhoduje [13]. Prevzaté z [13].

T-DeepCNN Systém T-DeepCNN je systém, ktorý sa zúčastnil konferencie ICDAR-2017 - *International Conference on Document Analysis and Recognition* [3], použitý na klasifikáciu typu písaného písma a datovanie dokumentov, je realizovaný prostredníctvom reziduálneho učenia a dávkovej normalizácie. Usporiadaná súťaž, ktorej sa systém zúčastnil, sa tiež zameriavala na klasifikáciu historických rukopisov z obdobia stredoveku, avšak ku klasifikácii podľa typu písaného písma pribudla úloha datovania dokumentov, pričom anotácie strán dokumentov k tejto úlohe boli v tvare intervalu. Tento systém dosahoval najlepšiu úspešnosť spomedzi všetkých systémov, ktoré sa súťaže zúčastnili [3].

Pre každú úlohu je použitá 50 vrstvová sieť architektúry ResNet. Siete sú natrénované na klasifikáciu výrezov z originálnych obrázkov pre každú úlohu zvlášť. Z obrázkov predstavujúce dokumenty sa extrahujú výrezy veľkosti 256×256 pixelov s hodnotou posunu 42×42 pixelov, ktorých veľkosť sa následne náhodne zmení [3]. Kvôli zlepšeniu výkonu systému sú aplikované transformácie na každý výrez vo fáze trénovania. Tieto transformácie pozostávajú z úpravy intenzity pridaním hodnoty získanej z normálneho rozdelenia. Anotácia každého výrezu taktiež predstavuje triedu strany dokumentu, z ktorej bol výrez extrahovaný [3].

Pri výslednej predikcii siete pre stranu dokumentu, sa farebné obrázky konvertujú na odtiene sivej a rukopisy sú rodzelené na prekrývajúce sa výrezy veľkosti 227×227 pixelov s hodnotou posunu 100×100 pixelov. Každý výrez je klasifikovaný zvlášť. Výslednú klasifikáciu potom predstavuje priemer predikcií všetkých výrezov dohromady [3]. Na dosiahnutie vačšej presnosti, sú predikcie tiež spriemerované cez súbor konvolučných neurónových sietí za použitia metódy *ensemble*. Obrázky sa navzorkovali vo viacerých mierkach. Pre *ensemble* predikcie každá konvolučná neurónová sieť vypočíta svoj výstup, ktorý je priemerom všetkých výrezov extrahovaných z obrázka v rámci jednej mierky. Následne tieto výstupy sú spriemerované cez všetky siete v súbore a vznikne jendotný výstup. Počet neurónových sietí v súbore bolo určených 5 [3].

PERO Martin Kišš, Jan Kohút, Karel Beneš a Michal Hradiš zrealizovali prístupy PERO opisované v článku *Importance of Textlines in Historical Document Classification* [17], s kto-

rými sa zúčastnili súťaže klasifikácia historických dokumentov, na konferencii ICDAR 2021. Usporiadaná súťaž pridala ďalšie dve úlohy k úlohám klasifikácia typu písaného písma a datovanie dokumentov. Pridané úlohy predstavujú klasifikáciu miesta vzniku a klasifikáciu skupiny fontu, v prípade dokumentov písaných tlačeným písmom. V prípade klasifikácie podľa typu fontu sa v poskytnutej dátovej sade vyskytovali prípady, kedy k obrázku predstavujúcemu stranu dokumentu bolo priradených viacero tried. Toto nastáva z dôvodu, že dokumenty sú často písané vo viacerých typoch fontov zároveň.

Realizované prístupy PERO riešia všetky spomenuté úlohy. PERO systémy sú založené na konvolučných neurónových sieťach, ktoré taktiež spracovávajú výrezy z originálnych obrázkov. Sú prezentované dva prístupy. Jeden pracuje nad výrezmi z obrázkov fixnej veľkosti, pričom druhý pracuje nad automaticky detekovanýnimi výrezmi riadkov obsahujúcich text [24].

Systém realizovaný nad textovými riadkami je postavený na architektúre VGGNet. Textové riadky sú automaticky detekovanými výrezmi riadkov obsahujúcich text. Na ich extrahovanie bol použitý systém analýzy rozloženia strán dokumentov z článku *Page Layout Analysis System for Unconstrained Historic Documents* [14], kde bol navrhnutý model založený na konvolučných neurónových sieťach, konkrétne architektúre ParseNet, na detekciu textových blokov. Detekované riadky majú rôznu dĺžku a výšku do veľkosti 30 pixelov. Varianta architektúry VGGNet, ktorú pre tento systém použili je VGG-16. Namiesto aktivačnej funkcie ReLu je však použitá aktivačná funkcia LeakyReLU vyjadrená naslednovne, kde a reprezentuje konštantu udávajúcu sklon a x predstavuje hodnotu, na ktorú sa aplikuje nelinearita [17]:

$$LeakyReLU(x) = \begin{cases} x & \text{x} > 0 \\ ax & \text{x} \leq 0 \end{cases}$$

Výstup neurónovej siete reprezentuje iba jeden výrez z obrázka a všetky takéto výstupy sú následne agregované do jedného konečného výstupu reprezentujúceho jednu celú stranu dokumentu [17]. Na agregáciu textových riadkov do jedného výstupu pre stranu dokumentu je použitých viacero stratégií. Navrhnuté stratégie su nasledovné: výber tej triedy, ktorú reprezentuje väčšina textových riadkov, spočítavanie pravdepodobností pre každú triedu a výber tej triedy, ku ktorej prislúcha najvyššia pravdepodobnosť, spriemerovanie resp. výpočet vektoru, ktorý predstavuje priemerné pravdepodobnosti pre jednotlivé triedy s výberom triedy, ktorej prislúcha najvyššia hodnota priemernej pravdepodobnosti [17]. Tieto stratégie sú použité v úlohách: klasifikácia podľa skupiny fontu, klasifikáciu podľa typu písaného písma a klasifikácia podľa miesta vzniku. Výpočet priemeru z individuálnych výsledkov a výpočet mediánu sú stratégie použité pri datovaní dokumentov [17].

Problém viacerých anotácií priradených k jednej strane dokumentu, v prípade klasifikácie podľa skupiny fontu, bol vyriešený agregovaním výstupov z funkcie krížová entropia - *cross entropy*. Najprv sa aplikuje krížová entropia pre všetky požadované výstupy, z ktorých sa následnej vypočítava celková chybovosť. Boli navrhnuté dva spôsoby na výpočet celkovej chybovosti, kde y_t je pravdepodobnosť priradená anotácii t a $CE(y,t)$ reprezentuje aplikáciu krížovej entropie medzi odozvou siete a požadovanou hodnotou [24]:

$$L_{min} = \min_t CE(y,t)$$

$$L_{avg} = \sum_t y_t CE(y,t)$$

K úlohe datovania dokumentov sa použila chybová funkcia predstavujúca modifikáciu funkcie *huber loss*, v rámci ktorej sa meria chybovosť v rozmedzí intervalu $< a, b >$, kde $m = (a + b)/2$ je stred a $r = (a - b)/2$ je polomer intervalu. Funkcia je matematicky vyjadrená nasledovne [24]:

$$L_{date}(y, a, b) = \begin{cases} a - y - r & y \geq a \\ a - y - r & y \geq a \\ (\frac{y-m}{r})^2 r & inak \end{cases}$$

Táto chybová funkcia rieši problém úlohy datovania dokumentov, ktorý vzniká tým, že k jednotlivým stranám dokumentu sú priradené intervaly období [17].

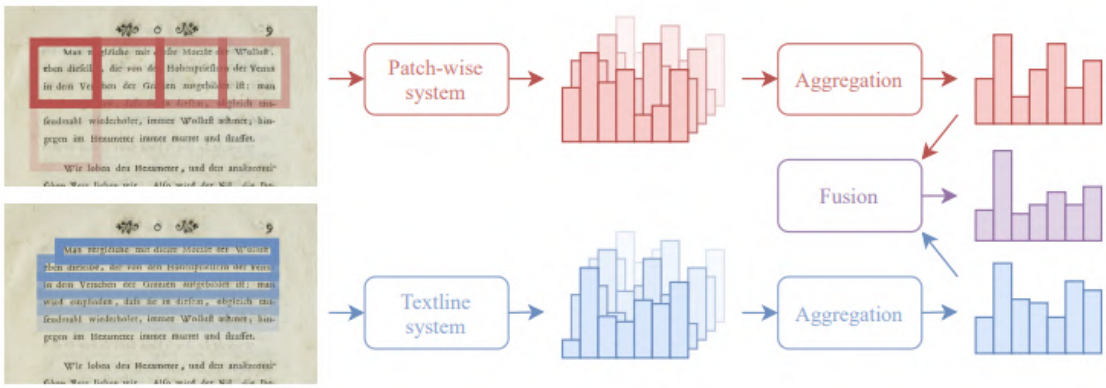
Systém realizovaný nad výrezmi fixnej veľkosti zo strán dokumentov, je druhá použitá metóda PERO systémov pre klasifikáciu historických dokumentov. Operuje nad neprekrývajúcimi sa výrezmi fixnej veľkosti 224×224 pixelov. Ide o neurónovú sieť architektúry ResNet-50 [17].

V rámci agregácie výstupu do jednotnej predikcie, je nad dátovou sadou realizovaná dátová augmentácia a každý obrázok z dátovej sady sa nachádza v štyroch rôznych mierkach v rámci ktorých je obrázok rozdelený na neprekrývajúce sa výrezy. Výstupy výrezov sa agregujú pre každú mierku zvlášť a agregácia štyroch mierok predstavuje výsledný výstup pre pôvodnú stranu dokumentu [17]. Výber výrezov, ktoré budú agregované je motivovaný konkrétnou úlohou. Prvá stratégia výberu výrezov je založená na výbere všetkých výrezov danej strany dokumentu a druhá je výber výrezov predstavujúcich textové regióny. Na detekovanie textových regiónov sa použil rovnaký systém, ako pri detekovaní textových riadkov [14]. Dodatočne boli obe stratégie skombinované a v tomto prípade je výstupom agregácia všetkých ôsmych mierok strany dokumentu. Tím predpokladal, že v prípade klasifikácie podľa typu fontu a písaného písma budú doležité výrezy obsahujúce text, ale aj mimotextové oblasti môžu obsahovať dôležité informácie v prípade úlohy lokalizácie dokumentov [24]. Táto kombinovaná stratégia bola po vykonaní rôznych experimentov, použitá v prípade klasifikácie podľa typu fontu, podľa typu písaného písma a podľa miesta vzniku, zatiaľ, čo v prípade datovania dokumentov bola použitá stratégia všetkých výrezov, nie len výrezov obsahujúcich text alebo kombinácia metód [17].

Pre klasifikáciu typov písma, bol najúspešnejšou metódou na výpočet celkového výstupu priemer výstupov všetkých mierok, pričom na úrovni výrezov sa vybralo 10 najspoľahlivejších výrezov, ktorých výstupy sa spriemerovali [24]. V prípade lokácie išlo o najspoľahlivejší výstup na úrovni mierok s rovnakým spôsobom agregácie výrezov, ako v predošlom prípade a v prípade datovania sa výstup na úrovni výrezov vypočítal ako medián všetkých výrezov a celkový výstup predstavuje priemer týcho jednotlivých výstupov cez všetky mierky [24].

Tieto dva spomenuté systémy sa následne pokúsili skombinovať prostredníctvom lineárnej interpolácie a logistickej regresie pre viacero tried [17].

Výsledky konajúcej sa súťaže ukázali, že systémy PERO sa umiestnili vo všetkých úlohách na prvých priečkach. Systémy založené na textových riadkoch boli úspešnejšie pri všetkých úlohách až na lokalizáciu dokumentov. V prípade určenia miesta vzniku dokumentu, bol systém pracujúci s neprekrývajúcimi sa výrezmy fixnej veľkosti úspešnejší. Víťazná však bola realizovaná fúzia spomínaných dvoch systémov. Úloha lokalizácie sa ukázala byť, ako najnáročnejšia, z dôvodu, že systémy na nej dosahovali najnižšie úspešnosti [24].



Obrázek 2.12: Fúzia oboch prístupov systémov PERO. Spracuje sa vstupná strana dokumentu systémom pracujúcim, s neprekrývajúcimi sa výrezmi fixnej veľkosti (červená) aj systémom pracujúcim s textovými riadkami (modrá). Výrezy a textové riadky sú agregované a sú nakoniec zlúčené do jediného rozhodnutia (fialová) [17]. Prevzaté z [17].

Kapitola 3

Dátové sady

K daným úlohám je k dispozícii viacero dátových sád zo spomínaných súťaží, ktoré sú v tejto kapitole bližšie popísané. Všetky historické dokumenty súčasťou týchto dátových sád pochádzajú z európskych digitálnych knižníc a sú datované približne medzi 9. a 18. storočím. Pre každú úlohu boli zverejnené aj testovacie dátové sady.

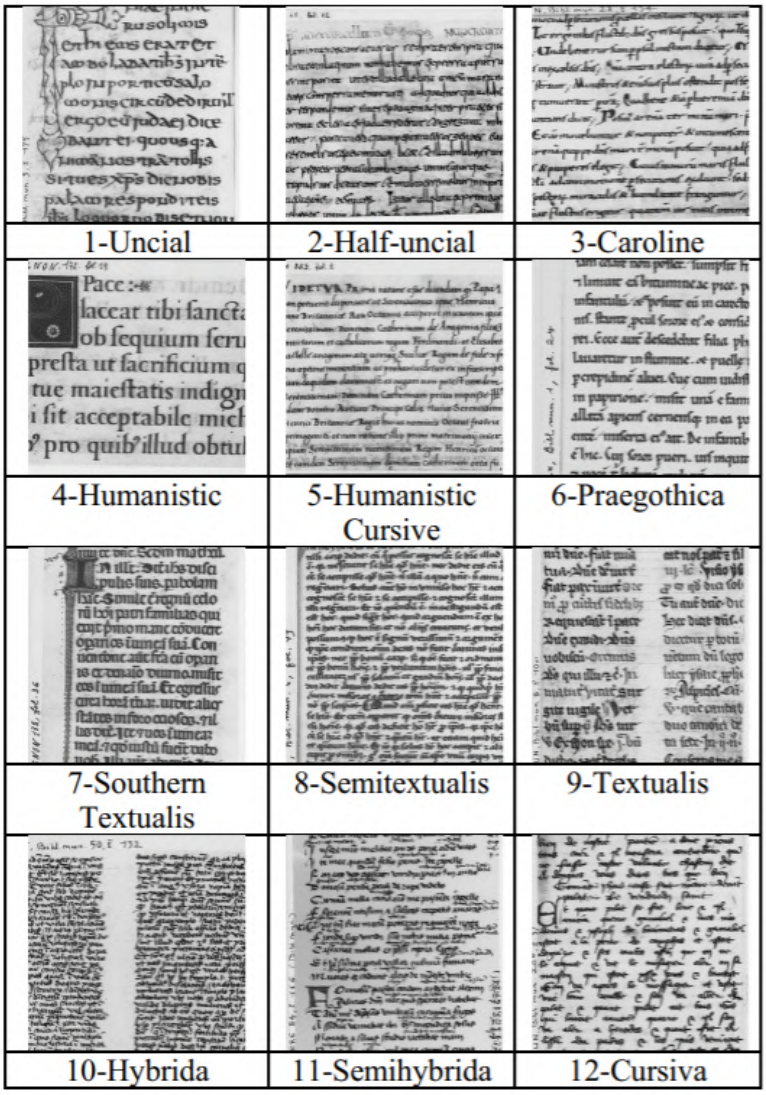
Dátová sada ICFHR 2016 - CLaMM

Nasledujúca dátová sada[2] bola poskytnutá zúčastneným súťaže na konferencii ICFHR 2016 [4]. Ide o sadu CLaMM - *Classification of Medieval Handwritings in Latin Script*, zameranej na klasifikáciu stredovekých rukopisov. Dátová sada predstavuje zbierku európskych rukopisov z obdobia stredoveku. Konkrétne ide o obrázky zachytávajúce dokumenty zo zbierky francúzskych katalógov, ktorých súčasťou je až 9800 dokumentov, digitálnej knižnice BVMM a Gallica. K dispozícii sú tri sady: trénovacia a dve testovacie. Všetky tri sady pozostávajú z bezfarebných obrázkov vo formáte TIFF veľkosti 100×150 pixelov, čo predstavuje časť nejakého rukopisu. Trénovacia sada obsahuje 2000 jednoznačne anotovaných obrázkov podľa typu písaného písma a dve testovacie sady obsahujú 1000 a 2000 obrázkov. Jedna z testovacích sád obsahuje jednoznačne anotované obrázky a druhá obsahuje obrázky, ktorým je priradených viacero anotácií na jednu stranu dokumentu, ktorý obrázok zachytáva [9]. Počet tried, ktoré reprezentujú typ písaného písma je 12, triedy sú znázornené na obrázku 3.1. K úlohe bola organizátormi súťaže zverejnená aj testovacia dátová sada.

Dátová sada ICDAR 2017

Dátová sada[3] CLaMM zo súťaže na konferencii ICFHR 2016 bola rozšírená v súťaži ICDAR 2017 [3], o obrázky anotované podľa obdobia vzniku, nakoľko datovanie dokumentov bolo novou úlohou v rámci konferencie. Tieto dáta taktiež pochádzajú zo zbierky francúzskych katalógov a digitálnej knižnice BVMM a Gallica. Trénovacia sada k úlohe datovania dokumentov je rovnako, ako dátová sada pre klasifikáciu rukopisov, zložená z bezfarebných obrázkov vo formáte TIFF o veľkosti 100×150, kde veľkosť sady je rovnako 2000 obrázkov [3]. Jedna anotácia priradená k jednému dokumentu predstavuje interval, teda najskoršie a najneskoršie možné obdobie vzniku. V dátovej sade existuje dohromady 15 takýchto možných intervalov datovania rukopisov od roku 1000 po rok 1600 [3]. Rovnako, ako pri určovaní typu písaného písma aj v tomto prípade sú k dispozícii dve testovacie sady, pričom jedna z

[2] https://clamm.irht.cnrs.fr/icfhr2016-clamm/data-set/
[3] https://clamm.irht.cnrs.fr/icdar-2017/data-set/



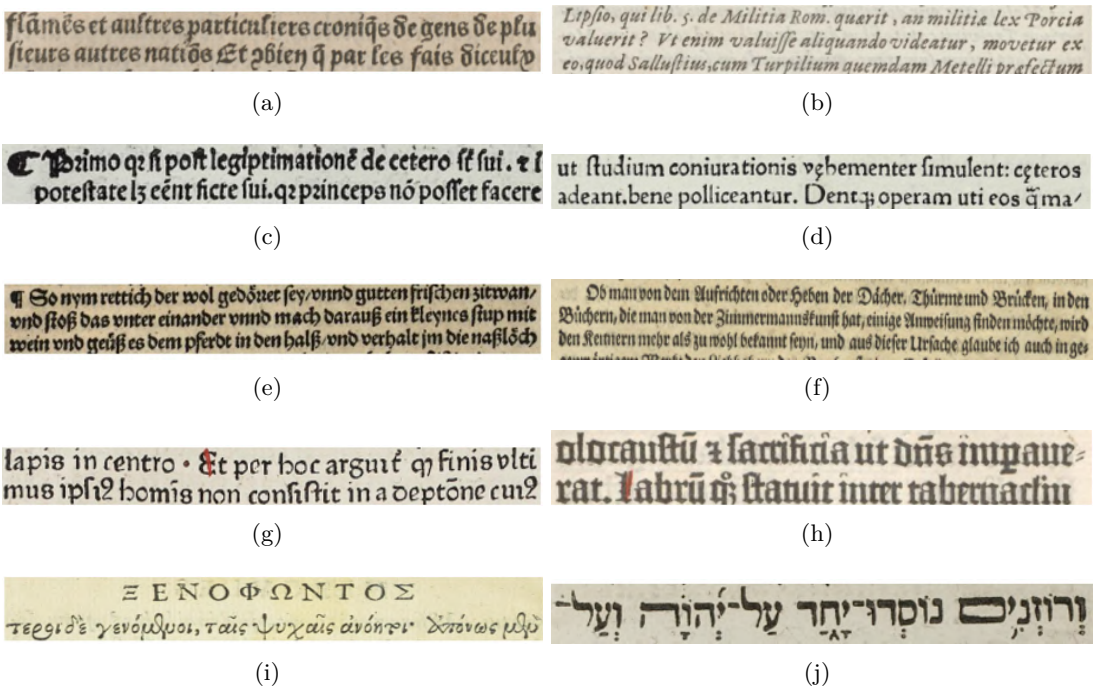
Obrázek 3.1: Príklady všetkých 12 tried typov písaného písma z dátovej sady zo súťaže ICFHR 2016. Prevzaté z [4].

nich má rovnaký formát, ako tá trénovacia s rovnakým počtom obrázkov a druhá obsahuje 1000 obrázkov v rôznych formátoch (JPG, TIFF) a v rôznych farebných prevedeniach a rozlíšeniach. Dátová sada k úlohe klasifikácia rukopisov je rovnaká, ako v ICFHR 2016 [3].

Dátová sada ICDAR 2021

V súťaži na konferencii ICDAR 2021 [24] boli úlohy rozšírené o klasifikáciu podľa skupiny fontu dokukmentov písaných tlačeným písmom a klasifikáciu miesta vzniku dokumentov. Konferencia teda pozostávala zo štyroch úloh: klasifikácia podľa písaného písma, klasifikácia podľa skupiny fontu, datovanie dokumentov a klasifikácia podľa miesta vzniku [24].

Dátová sada dokumentov písaných tlačeným písmom, bola zverejnená v článku *Dataset of Pages from Early Printed Books with Multiple Font Groups* [23]. Sada pozostáva z celkovo 35 623 obrázkov historických dokumentov rôznych rozlíšení a pochádza z niekoľkých digi-



Obrázek 3.2: Príklady všetkých 10 tried skupiny fontov z dátovej sady ICDAR 2021. (a) Bastarda (b) Italic. (c) Rotunda. (d) Antiqua. (e) Schwabacher. (f) Fraktur. (g) Gotico-Antiqua. (h) Textura. (i) Greek. (j) Hebrew. Prevzaté z [21].

tálnych knižníc. Veľkú časť svojho digitalizovaného materiálu z 15. až 18. storočia poskytli knižnice v Berlíne, Erlangene, Göttingene, Mníchove a Stuttgarte [23]. Aby sa vyvážilo vysoké zastúpenie dokumentov zo 17. a 18. storočia, knižnice v Londýne, Kölne a Heidelbergu poskytli dokumenty z 15. storočia. Podklady ku skupine fontov *Greek* a *Hebrew* boli získané od digitálnej knižnice *Herzog August Bibliothek Wolfenbüttel 40*. Každý jeden obrázok dokumentu má priradených 1 až 5 anotácií reprezentujúcich typ fontu, pričom celkový počet typov je 10. Počet tried je však 12, zvyšné dve predstavujú kategóriu *iné*, ak ide o font, ktorý neprislúcha ani jednej z tried a kategóriu *nie je font*, v prípade ak sa nejedná o žiadny typ fontu. Obrázky sú anotované typom fontu, ktorý sa používa v dokumente najviac, niektoré stránky však majú viacero hlavných typov [21]. Testovacia dátová sada[4] obsahuje celkovo 5506 obrázkov, pričom z nich 2753 je originálnych a zvyšné vznikli po dátovej augmentácii, konkrétne zmenami veľkostí, aby sa preverila robustnosť jednotlivých systémov [23].

Druhou novou úlohou bola klasifikácia dokumentov podľa miesta vzniku. Pre túto úlohu je dostupná aj validačná sada. Dáta k tejto úlohe pochádzajú z viacerých zdrojov. Ide o katalógy obsahujúce rukopisy vydávané vo Francúzsku konvertované do XML súborov[5], špecifické paleografické štúdie na tvorbu kníh a materiály z prebiahjúceho projektu ohľadne *Saint-Bertin*[6]. Počet tried je 13 a ide o vyše 5000 obrázkov. Triedy lokácií boli zvolené tak, aby reprezentovali vysoko zaľudnené a veľké miesta, či už mestá alebo celé regióny. [23].

V rámci tejto súťaže bola poskytnutá nová trénovacia a testovacia dátová sada slúžiaca k datovaniu dokumentov. Tieto sady pochádzajú z *e-codies: Virtual Manuscript Library of*

[4] https://zenodo.org/record/4836551
[5] https://github.com/oriflamms/CMDF
[6] https://saint-bertin.irht.cnrs.fr

(a) (b)

(c)

Obrázek 3.3: Ukážky historických dokumentov z dátovej sady ICDAR 2021.

Switzerland[7], čo je švajčiarska historická digitálna knižnica rukopisov. Forma anotácie je rovnaká ako pri predchádzajúcej sade, teda k obrázkom je priradený interval, pričom sa posudzovali iba tie, ktoré boli jednoznačne určené, čo znamenalo, ak sa názory expertov nelíšili vo vyše 50 rokoch. Brali sa do úvahy dokumenty, ktoré neboli datované skôr ako obdobie 9. storočia a neskôr ako 17. storočie. Požiadavkam vyhovovalo celkovo 1698 dokumentov. Ide o trénovaciu aj testovaciu sadu súčasne. Z týchto dokumentov boli vybrané náhodné obrázky reprezentujúce strany dokumentu a trénovacia a testovacia sada[8] takto obsahuje celkovo 10 294 a 2516 obrázkov [23].

[7] http://e-codices.unifr.ch/en
[8] https://zenodo.org/record/4836687

Kapitola 4

Návrh riešenia

V tejto kapitole sa vyhodnocuje záver analýzy vyplývajúcej z existujúcich metód klasifikácie historických dokumentov opísaných v kapitole 2. Na základe tohto vyhodnotenia, sú navrhnuté možné riešenia, ktoré táto práca realizuje. Riešenia sú zamerané konkrétne na úlohu klasifikácia historických dokumentov podľa miesta vzniku, nakoľko systémy pri tejto úlohe dosahovali najhoršie výsledky.

4.1 Vyhodnotenie analýzy existujúcich metód

Z analýzy existujúcich riešení opísaných v kapitole 2.5, vyplynulo, že pri úlohách klasifikácia podľa typu fontu, typu písaného písma a pri datovaní dokumentov, nesú textové riadky dôležitejšiu informáciu, ako pri úlohe klasifikácia podľa miesta vzniku [24]. Pri týchto úlohách dosahovali výrazne lepšie výsledky systémy, ktoré boli postavené na automatickej detekcii riadkov textu z dokumentu. Na túto informáciu poukázali systémy PERO v súťaži na konferencii ICDAR 2021 [24]. Na základe spomínaných výsledkov súťaže sa teda pri lokalizácii historických dokumentov predpokladá, že aj mimotextové oblasti dokumentu dôrazne prispievajú k rozhodovaniu siete [24]. Vo väčšine prípadov v rámci konajúcej sa súťaže boli najúspešnejšie systémy PERO, ktoré kombinovali prístup nad fixnými výrezmi s prístupom nad textovými riadkami [24].

Na základe analýzy je najnáročnejšou úlohou klasifikovať dokumenty podľa miesta vzniku. Pri tejto úlohe, existujúce systémy dosahovali najhoršie výsledky, naopak najlepšie výsledky dosahovali pri klasifikácii podľa typu fontu a písaného písma [17].

Čo sa týka použitých architektúr, väčšina prístupov bola založená na konvolučných neurónových sieťach. Išlo hlavne o siete architektúry VGGNet a ResNet [24].

4.2 Návrhy riešení

Východzajúc z toho, že úloha lokalizovania historických dokumentov sa ukázala byť najnáročnejšou, sa návrhy riešení budú týkať len tejto úlohy. V nasledujúcej časti sú opisané jednotlivé návrhy možných riešení.

Spracovanie obrázkov Nakoľko sa ukázalo, že pri klasifikovaní dokumentov na základe miesta vzniku, je cennou informáciou aj pozadie dokumentu, nie len samotný text [24], by nebolo vhodné sa sústreďovať len na oblasti v ktorých sa vyskytuje text. Ukázalo sa, že kombinácia textových riadkov s výrezmi fixnej veľkosti je relatívne úspešná. Z tohto dôvodu

by vstupom do siete mohli byť výrezy fixnej veľkosti v tvare obdĺžnika, narozdiel od väčšiny systémov, ktoré pracovali na výrezoch v tvare štvorcov. Každý obrázok, by sa teda rozdelil na neprekrývajúce sa obdĺžniky fixnej veľkosti, ktoré by mohli imitovať textové riadky, ale zároveň by sa do siete dostala aj informácia mimo textu, nakoľko by sa brali do úvahy všetky výrezy extrahované z obrázka. Vstup do siete by teda predstavoval jeden výrez z pôvodného obrázka zachýtavajúceho jednu stranu dokumentu, rovnako ako v prípade opisovaných systémov v kapitole 2.5. Trénovanie siete by prebiehalo na úrovni výrezov, pričom pri inferencii by sa výstup siete pre každý výrez pôvodného obrázka následne agregoval do jednotnej predikcie.

Konvolučná neurónová sieť Na klasifikáciu spomínaných výrezov v tvare obdĺžnika by sa mohla použiť klasická hlboká konvolučná neurónová sieť, ktorá by mala na vstupe jeden výrez obrázka. Mohlo by ísť o architektúru ResNet opisovanú v kapitole 2.2. Išlo by teda o reziduálne bloky s konvolučnými vrstvami a vrstvami združovania podľa maxima.

Pri klasifikácii výrezov pomocou tejto konvolučnej neurónovej siete, by sa namiesto vrstvy združovania podľa priemeru, ktorá sa nachádza pred plne prepojenou vrstvou, mohol použiť mechanizmus *self-attention*. Tento mechanizmus by zabezpečil, aby sa do výsledného agregovaného vektora, ktorý slúži ako vstup do plne prepojenej vrstvy, dostali iba relevantné informácie. Pre každý vektor výstupného tenzoru posledného reziduálneho bloku, by sa vypočítali *attention* váhy a následne by sa tieto vektory váhovane sčítali prostredníctvom týchto váh. Po plne prepojenej vrstve by sa aplikovala funkcia *softmax*, ktorá by vrátila jednotlivé pravdepodobnosti tried pre každý výrez.

Vychádzajúc z vizualizácii systému DeepScript [4], ktorý je opísaný v kapitole 2.5, by tento výsledný agregovaný vektor po váhovanom sčítaní taktiež mohol slúžiť ako informácia o tom, na základe čoho sa daná sieť najviac rozhoduje. Táto informácia by sa následne mohla podobným spôsobom vizualizovať.

Pseudo labeling Klasická konvolučná neurónová sieť, prípadne sieť opisovaná v predošlej časti s použitím mechanizmu *self-attention*, by sa mohla použiť k tzv. semi-supervizovaným experimentom. Nakoľko existuje viacero dátových sád, k natrénovaniu siete by sa mohli využiť aj iné obrázky historických dokumentov, z existujúcich dátových sád z kapitoly 3, ktoré nie sú anotované. Natrénovaná konvolučná neurónová sieť by sa mohla použiť k anotácii výrezov z týchto obrázkov a tieto novo anotované výrezy, by sa následne stali súčasťou trénovacej sady, na ktorej by bola sieť znova natrénovaná.

Anotácia týchto obrázkov by mohla prebiehať rôznymi možnými spôsobmi. Prichádza do úvahy výber n výrezov s najväčšou pravdepodobnosťou predikovanej triedy pre každý obrázok zvlášť. Týmto výrezom by sa priradili triedy predikované modelom a stali by sa súčasťou novej trénovacej sady. Ďalší spôsob by mohol predstavovať výber n výrezov pre každý obrázok, pre ktoré by sa predikcie modelu spriemerovali a tento priemer predikcií by predstavoval triedu pridelenú pôvodnému obrázku, ktorý by sa následne spracoval rovnako, ako pôvodná dátová sada, teda rozdelil na výrezy.

Táto opísaná metóda je súčasťou semi-supervizovaného učenia a nazýva sa *pseudo-labeling*, keďže pôvodným obrázkom, ktoré nie sú anotované sa pridelia tzv. pseudo anotácie [1]. Rozšírenie dátovej sady by mohlo dopomôcť k zlepšeniu úspešnosti konvolučnej neurónovej siete.

Masked image modelling Ďalšou možnosťou je použiť model BEiT postavený na vizuálnom transformeri, opisovaný v kapitole 2.4. Motiváciou je, že väčšina existujúcich prístupov

používa len konvolučné neurónové siete a množstvo dostupných dát. Išlo by teda o predtrénovanie transformera prostredníctvom metódy modelovanie maskovaných obrázkov, čo predstavuje ďalšiu formu semi-supervizovaného učenia. Trénovanie vizuálnych transformerov si vyžaduje mnoho dát a preto v rámci tohto prístupu, by sa taktiež mohli na predtrénovanie využiť aj historické dokumenty mimo dátovej sady určenej k úlohe lokalizácie.

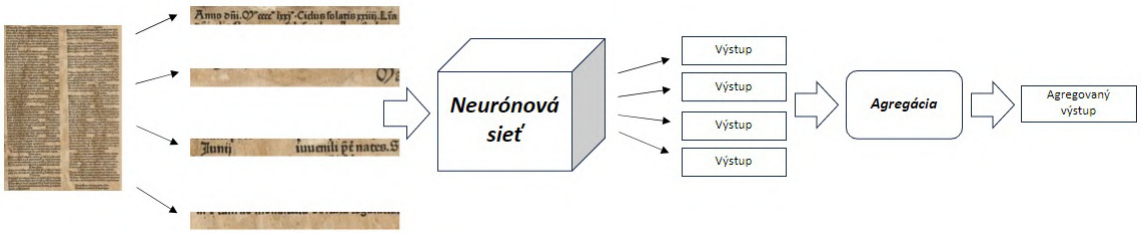
Vstupné obrázky (v tomto prípade výrezy pôvodnej strany dokumentu), by sa rozdelili na menšie výrezy fixnej veľkosti. Istá časť výrezov by sa náhodne zamaskovala využitím algoritmu blokového maskovania. Výrezy by sa vektorizovali a následnou lineárnou transformáciou by k vzniknutým embedingom výrezov bol pridaný pozičný embeding a tzv. klasifikačný embeding rovnako ako to bolo opisované v spomínanej kapitole. Táto sekvencia by predstavovala vstup do daného enkódera transfomeru. Mohol by sa použiť klasický enkóder, teda súbor vrstiev, pričom každá vrstva by bola zložená z *multihead self-attention* a MLP blokov založená na reziduálnom učení [2].

Na generovanie diskrétnych vizuálnych tokenov z pôvodných obrázok by sa mohol aplikovať prístup vychádzajúci z navrhnutého modelu HuBERT spomenutého v 2.4. V prístupe HuBERT používajú zhlukovací algoritmus *k-means* na extrahované príznaky. Tu by sa mohla použiť natrénovaná konvolučná neurónová sieť, jej výstup z posledného konvolučného bloku by sa transformoval na diskrétne vizuálne tokeny prostredníctvom algoritmu *k-means*. Vizuálne tokeny by teda predstavovali diskrétne hodnoty, ktoré sa vytvorili prostredníctvom vzniku zhlukov.Sieť by následne predikovala tokeny pôvodného nepoškodeného obrázka [2].

Po predsňtrénovaní transfomera, by sa následne pripojila klasifikačná vrstva - združovanie podľa priemeru s lineárnou vrstvou a funkciou *softmax* a transformer by sa natrénoval na úlohu klasifikácie historických dokumentov na základe miesta vzniku.

Agregácia výstupov siete Výsledná predikcia siete, by mala prebiehať na úrovni obrázkov zachytávajúcich celé strany dokumentu a nie na výrezoch. Nakoľko sa neuronová sieť trénuje na výrezoch z pôvodných obrázkov, je potrebné navrhnúť spôsob agregácie výstupov siete prislúchajúcich k výrezom. Systémy PERO a T-DeepCNN opísané v kapitole 2.5, na agregovanie výstupov siete do jednotnej predikcie používajú metódu *ensamble*. Obrázok sa vyhodnocuje vo viacerých rozlíšeniach (mierkach) a následne sa tieto predikcie taktiež agregujú [3]. Tento prístup by sa taktiež mohol vyskúšať v tejto práci, nakoľko vo väčšine prípadov majú obrázky v dátovej sade rôzne rozlíšenia. Pre každé rozlíšenie, by sa predikcie mohli následne spriemerovať a získala by sa finálna predikcia pre daný obrázok.

Pri agregácii výstupu na úrovni výrezov je možné postupovať viacerými spôsobmi. Prvý spôsob by mohol predstavovať výber n výrezov s najväčšou pravdepodobnosťou predikovanej triedy, ktorých predikcie modelom sa následne spriemerujú. Ďalej by mohlo ísť o výber výrezu s predikciou najväčšej pravdepodobnosti a trieda predikovaná modelom pre tento výrez by bola výslednou predikciou pre celý obrázok. Možné by bolo aj spriemerovanie predikcií modelu pre všetky výrezy. Podobné spôsoby agregácie výstupov realizovali aj existujúce riešenia [24]. Obrázok 4.1 odzrkadluje navrhované riešenie.



Obrázek 4.1: Ukážka inferencie siete na úrovni strán dokumentov. Obrázok sa rozdelí na neprekrývajúce sa výrezy v tvare obdĺžnika. Na vstupe neurónovej siete je jeden výrez. Pre každý výrez, výstup sieťe predstavuje jednotlivé pravdepodobnosti tried, najpravdepodobnejšie triedy, ktoré predstavujú predikcie pre dané výrezy sa následne agregujú do jednotnej predikcie, ktorá bude predstavovať predikovanú triedu pôvodnej strany dokumentu.

Kapitola 5

Implementácia

V tejto kapitole sú opísané riešenia, ktoré boli v práci implementované a vychádzajú z návrhu v kapitole 4, pre úlohu klasifikácia historických dokumentov podľa miesta vzniku. Taktiež opisuje niektoré implementačné detajly riešení. Prvá časť sa venuje opisu použitých technológií pri implementácii, následne je opísaný spôsob predspracovania dátovej sady. V ďalších kapitolách sú opísané implementačné detajly riešení.

5.1 Použité nástroje

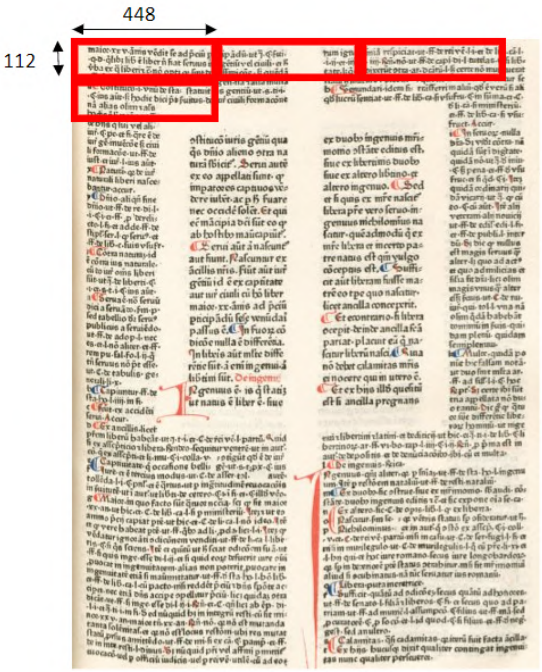
Hlavným programovacím jazykom tejto práce je Python, pričom na trénovanie neurónových sietí je použitá knižnica PyTorch[9]. Táto knižnica podporuje vývoj a trénovanie pokročilých hlbokých neurónových sietí. Jej výhody sú jednoduchosť, flexibilita, efektívne využívanie pamäte a dynamické výpočtové grafy, taktiež podpora pre GPU [12].

5.2 Predspracovanie dátovej sady

Kvôli vysokému rozlíšeniu niektorých obrázkov je trénovacia aj validačná sada predspracovaná vopred pomocou krátkeho skriptu. Obrázky sú načítavané prostredníctvom knižnice OpenCV[10], ide o knižnicu navrhnutú na riešenie problémov počítačového videnia. Obrázky sa rozdelia na neprekrývajúce sa výrezy veľkosti 112×448 pixelov, pri hodnote posunu 448 pixelov na šírku a 112 pixelov na výšku. V prípade, ak rozmery pôvodného obrázka nie sú deliteľné týmito hodnotami, zvyšené výrezy, ktorých rozmery nekorešpondujú s hodnotou 112×448 , nie sú zahrnuté do dátovej sady. Tento proces je znázornený na obrázku 5.1. Tieto rozmery boli určené z dôvodu, že väčšina konvolučných neurónových sietí, pracuje nad obrázkami fixnej veľkosti 224×224 pixelov. Je teda zachovaná veľkosť plochy výrezov a mení sa len tvar. Obrázky sú ponechané v RGB. Príklady výrezov po spracovaní obrázkov je možné vidieť na obrázku 5.2.

Po spracovaní všetkých obrázkov z pôvodnej trénovacej a validačnej sady, teda vzniká nová trénovacia a validačná sada obsahujúca výrezy uložené pod názvom v tvare: `[pôvodný názov obrázka]#[číslo výrezu]`.

[9] https://pytorch.org
[10] https://docs.opencv.org/4.x/index.html



Obrázek 5.1: Spôsob predspracovania obrázkov. Červené obdĺžniky predstavujú novovzniknuté výrezy veľkosti 112×448 pixelov, ktoré vznikli rozdelením pôvodného obrázka.

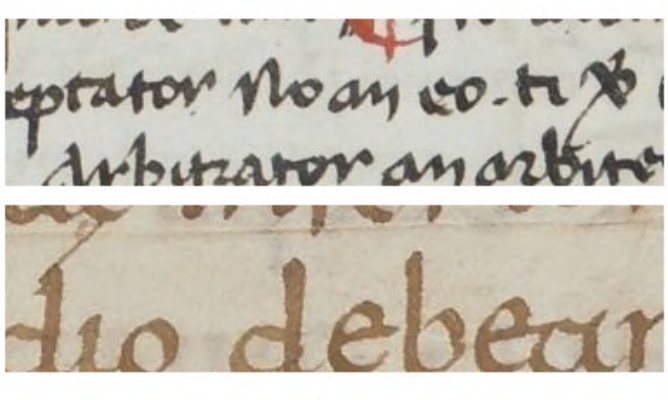
5.3 Konvolučné neurónové siete

Architektúra konvolučnej neurónovej siete, ResNet-50[11], ktorá je v práci použitá, je súčasťou knižnice PyTorch, konkrétne modulu torchvision. Tento modul obsahuje rôzne predtrénované modely na riešenie problémov počítačového videnia. Architektúra je opísaná v kapitole 2.2. Ide o predtrénovanú sieť, takže boli použité najaktuálnejšie váhy predtrénované na dátovej sade ImageNet.

Vzhľadom k tomu, že táto sieť je trénovaná na klasifikáciu do 1000 tried na obrázkoch veľkosti 224×224 pixelov, bola potrebná dodatočná modifikácia lineárnej vrstvy siete, aby jej výstupom bol vektor pravdepodobností pre 13 tried.

Attention Súčasťou použitej siete je pred lineárnou vrstvou, vrstva združovania podľa priemeru - *global average pooling*. Rozmery matice predstavujúcej výstup z konvolučných blokov sú $N \times C \times H \times W$, kde N predstavuje veľkosť trénovacej dávky - *batch*, C je počet kanálov, H a W sú rozmery, výška a šírka. Keďže v práci je konvolučná sieť trénovaná na výrezoch veľkosti 112×448 pixelov, rozmery výstupu posledného konvolučného bloku sú nasledovné: $2048 \times 4 \times 14$. Ako je spomenuté v kapitole 2.2, *global average pooling* vrstva slúži k tomu, aby sa vektory výstupnej mapy z konvolučných blokov agregovali do matice veľkosti $N \times C$, ktorá následne bude predstavovať vstup do plne prepojenej vrstvy. Agregácia touto metódou spočíva v spriemerovaní vektorov cez H a W. Vychádzajúc z návrhu namiesto vrstvy združovania podľa priemeru, je implementovaný mechanizmus *self-attention* v triede `SelfAttentionPooling`. Vektory tohto výstupu cez rozmery H a W sú vstupom do lineárnej vrstvy, pričom výstupy tejto lineárnej vrstvy predstavujú *attention*

[11] https://github.com/pytorch/vision/blob/main/torchvision/models/resnet.py



Obrázek 5.2: Ukážka výrezov zo spracovanej dátovej sady.

váhy. Každéme vektoru je teda priradená nejaká váha. Tieto váhy sa normalizujú použitím funkcie *softmax*, a prostredníctvom týchto normalizovaných váh sa jednotlivé vektory cez H a W váhovane sčítajú a vznikne výsledná matica v tvare $N \times C$, ktorá je následne v stupom do plne prepojenej vrstvy. Na výstup plne prepojenej vrstvy sa aplikuje funkcia *softmax* a získavajú sa pravdepodobnosti pre všetkých 13 tried klasifikácie.

5.4 Semi-supervizované učenie

Pseudo labeling Predtrénovaná konvolučná neurónová sieť architektúry ResNet-50, ktorá je súčasťou knižnice PyTorch a je dotrénovaná na spracovanej dátovej sade na úlohu klasifikácia podľa miesta vzniku, je následne použitá k anotácii zvyšných obrázkov historických dokumentov v dátovej sade, určených pre iné úlohy. Anotácia týchto obrázkov prebieha nasledovne: obrázky sa spracujú na výrezy a vyberie sa 10% výrezov, pre ktoré majú výstupy siete najvyššiu pravdepodobnosť. Tieto výrezy sa pridali do pôvodnej trénovacej dátovej sady a boli k nim priradené anotácie, ktoré predstavujú predikcie natrénovaného modelu.

Masked image modelling K predtrénovaniu na úlohu modelovanie maskovaných obrázkov je v práci využitý model BEiT (viď kapitolu 4.2), ktorý je súčasťou PyTorch knižnice transformers. Moduly, ktoré sa používajú sú nasledovné: `BeitConfig`, ktorý slúži ku konfigurácii modelu, `BeitFeatureExtractor`, ktorý zabezpečuje transformáciu vstupných obrázkov, `BeitForMaskedImageModeling`, ktorý sa použil k predtrénovacej úlohe modelovania maskovaných obrázkov a trieda `BeitForImageClassification` sa následne využila pri dolaďovaní modelu na klasifikačnú úlohu lokalizácie. Model BEiT je podrobne opísaný v kapitole 2.4.

K tokenizácii výrezov sa implementovalo riešenie vychádzajúce z návrhu a teda predtrénovaná sieť ResNet-50 sa použila ako tokenizér. Na podvzorkovaný výstup z posledného konvolučného bloku sa aplikovala metóda *k-means*, ktorá je súčasťou knižnice OpenCV. Po tejto operácii vznikne vizuálny token, ktorý sa pri trénovaní porovnáva s modelom predikovanými vizuálnymi tokenmi maskovaného vstupu.

Po predtrénovaní sa použil modul `BeitForImageClassification` k natrénovaniu na úlohu klasifikácia dokumentov podľa miesta vzniku. Lineárna vrstva bola upravená, aby výstupy predstavovali pravdepodobnosti pre 13 tried.

5.5 Agregácia výstupu siete

Agregácia výstupov siete na úrovni výrezov je v práci implementovaná nasledovným spôsobom: na základe výstupov siete pre výrezy vzniknuté z pôvodného obrázka (viď 5.1), sa vyberie 10% výrezov zo všetkých výrezov s najistejšou predikciou triedy. Výslednú predikovanú triedu predstavuje priemer predikcií siete pre vybrané výrezy.

Keďže obrázky v dátovej sade sú v rôznych rozlíšeniach, je v práci implementovaná metóda *ensamble* podľa návrhu v kapitole 4.2. Obrázok sa teda vyhodnocuje v piatich fixne stanovených mierkach, zároveň pre každú jednu mierku sa agregujú výstupy na úrovni výrezov a následne sa výstupy všetkých piatich sietí (každá prislúchajúca jednej mierke), spriemerujú. Tento priemer reprezentuje celkový výstup siete pre jeden vstupný obrázok.

Kapitola 6

Experimenty a výsledky

Nasledujúca kapitola sa venuje prevedeným experimentom a dosiahnutým výsledkom. Prevedené experimenty sa dajú rozdeliť do dvoch hlavných kategórií. Prvou kategóriou sú experimenty, ku ktorým sa použila len časť dátovej sady, teda sada určená konkrétne k úlohe klasifikácia dokumentov podľa miesta vzniku. Druhá kategória zahŕňa experimenty nad celou dátovou sadou. Tieto experimenty sú bližšie opísané v nasledujúcich častiach kapitoly. Riešenie, ktoré dosahovalo najlepšie výsledky na validačnej sade bolo následne evaluované na testovacej sade poskytnutej organizátormi súťaže.

Spôsob vyhodnocovania K vyhodnoteniu všetkých experimentov sa použila metrika *accuracy*, ktorá sa počíta nasledovným spôsobom: `počet správne klasifikovaných obrázkov / celkový počet obrázkov`. Vyhodnocovanie prebiehalo na dvoch úrovniach, na úrovni výrezov pôvodných obrázkov a na úrovni stránok, čiže celých obrázkov.

6.1 Dátová sada

Dátová sada, ktorá bola použitá v tejto práci pochádza z medzinárodnej konferencie ICDAR 2021 [24] a bola poskytnutá všetkým zúčastneným súťaže. Ako už bolo spomenuté v kapitole 3, konferencia sa zameriavala na štyri úlohy: klasifikácia podľa typu fontu, klasifikácia podľa typu písaného písma, datovanie dokumentu a klasifikácia podľa miesta vzniku. Dátová sada je zložená teda zo štyroch častí a každá časť zodpovedá jednej úlohe. Pre každú úlohu boli zverejnené aj testovacie dátové sady, pričom pre úlohu klasifikácia podľa miesta vzniku bola dodaná aj validačná sada.

Typ fontu Trénovacia dátová sada pre úlohu klasifikácia skupiny fontu bola zverejnená v článku *Dataset of Pages from Early Printed Books with Multiple Font Groups* [22]. Celkový počet obrázkov v tejto dátovej sade je 35623. Každý obrázok je anotovaný. Anotácií k jednému obrázku môže byť 1 až 5, nakoľko niekoľko obrázkov v dátovej sade obsahuje obrázky dokumentov písané vo viacerých typoch fontov. Tabuľka 6.1 znázorňuje počet obrázkov vhzľadom na pridelený počet fontov. Dátová sada obsahuje 12 tried. 10 tried predstavuje konkrétne fonty, ktoré sú znázornené na obrázku 3.2. Zvyšné triedy predstavujú kategóriu *Iné* a kategóriu *Nie je font*. Podiel obrázkov k jednotlivým triedam zachytáva tabuľka 6.2.

Typ rukopisu Trénovacia dátová sada pre úlohu klasifikácia podľa typu písaného písma bola poskytnutá v medzinárodnej konferencii ICDAR 2017 [3], ktorá vychádza z pôvodnej konferencie ICDAR 2016. Súčasná trénovacia sada pre túto úlohu zahŕňa pôvodnú trénovaciu a dve testovacie. Celkový počet obrázkov v tejto dátovej sade je 6540. Ku každému

Počet anotácií na obrázok	Počet obrázkov
1	30855
2	3732
3	891
4	136
5	9

Tabulka 6.1: Počet obrázkov vzhľadom na počet anotácií v trénovacej sade, pre úlohu klasifikácia podľa skupiny fontu.

Typ fontu	Počet obrázkov	Podiel obrázkov
Antiqua	8018	22.5%
Bastarda	974	2.7%
Fraktur	7333	20.6%
Gotico-Antiuqua	2589	7.3%
Greek	507	1.4%
Hebrew	1046	2.9%
Italic	2887	8.1%
Rotunda	5088	14.3%
Schwabacher	2640	7.4%
Textura	1293	3.6%
Iné	1470	4.1%
Nie je font	7734	21.7%

Tabulka 6.2: Počet obrázkov a celkový podiel pre každý typ fontu v trénovacej sade.

obrázku je priradená práve jedna anotácia. Počet tried je 12. Každá z tried predstavuje určitý typ písaného písma, jednotlivé triedy sú znázornené na obrázku 3.1. Podiel obrázkov k jednotlivým triedam je znázornený v tabuľke 6.3.

Obdobie vzniku Ako už bolo spomenuté v kapitole 3, z pôvodných metadát virtuálnej švajčiarskej knižnice, sa použili údaje *„Od“* a *„Do“*, ako anotácie dokumentov. Počet dokumentov, ktoré vyhovovali kritériam je 1698. Ide o dokumenty, ktoré sú v rozmedzí od 9. po 17 storočie. Z každého dokumentu bolo získaných niekoľko náhodných stránok [24]. Tieto stránky predstavujú obrázky v dátovej sade, ktorých celkový počet je 10294. Podiel obrázkov podľa storočí je znázornený v tabuľke 6.4.

Miesto vzniku Táto sada pochádza z niekoľkých zdrojov. Zdroje a tvorba dátovej sady sú bližšie popísané v kapitole 3. Ide o 13 tried, ktoré reprezentujú väčšie, zaľudnené oblasti. Celkový počet obrázkov v trénovacej sade je 5517. K úlohe bola dodaná aj validačná sada so 64 obrázkami. Zastúpenie tried v trénovacej, validačnej a testovacej sade je znázornené v tabuľke 6.5.

Ako už bolo spomenuté v prvej časti tejto kapitoly, v závislosti od experimentu, sa používa len časť trénovacej sady, teda určenej pre úlohu lokalizácie alebo celá sada pre všetky úlohy. Pre túto úlohu je k dispozícii aj fixná validačná dátová sada, ktorá je použitá na validáciu.

Typ rukopisu	Počet obrázkov	Podiel obrázkov
Uncial	409	6%
Half-uncial	504	8%
Caroline	452	7%
Humanistic	207	6%
Humanistic Cursive	299	5%
Praegothica	441	7%
Southern Textualis	791	12.1%
Semitextualis	324	5%
Textualis	404	6%
Hybrida	902	13.8%
Semihybrida	635	10%
Cursiva	703	10.7%

Tabulka 6.3: Počet obrázkov a celkový podiel pre každý typ rukopisu v trénovacej sade.

Storočie	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII
Podiel obrázkov	15.3%	4%	6%	9%	7%	12.1%	32.3%	9%	4%

Tabulka 6.4: Celkový podiel obrázkov pre každé storočie v trénovacej sade pre úlohu datovania dokumentov.

Typ rukopisu	Trénovacia sada	Validačná sada	Testovacia sada
Cluny	180	4	25
Corbie	795	5	25
Citaeux	140	5	25
Florence	45	5	25
Fontenay	250	5	25
Himanis (Paris chancery)	230	5	25
Milan	55	5	25
Mont-Saint-Michel	182	5	25
Paris (book trade)	467	5	25
Saint-Bertin	2988	5	25
Saint-Germain-des-Prés	55	5	25
Saint-Martial-de-Limoges	75	5	25

Tabulka 6.5: Zastúpenie tried pre úlohu lokalizácie dokumentu v trénovacej, validačnej a testovacej sade.

Vychádzajúc z článku [17] a nakoľko výsledky pri trénovaní sa na validačnej sade javili pomerne horšie, sa v rámci experimentov vyskúšala tzv. krížová validácia. Tzn., že bola vytvorená nová validačná dátová sada, ktorá pochádza z pôvodnej trénovacej s rovnakým zastúpením tried. Pôvodná validačná sada sa tým pádom stala súčasťou trénovacej sady. Všetky experimenty sa teda vykonávali na pôvodných aj novovzniknutých sadách zároveň.

Testovacia sada[12] bola poskytnutá a zverejnená organizátormi súťaže. Táto sada sa použila na otestovanie najúspešnejšieho riešenia, teda riešenie s najvyššou presnosťou na validačnej sade. Ako už bolo zmietené v kapitole 5, trénovacia a validačná sada sú predspracované na výrezy fixnej veľkosti. Trénovanie teda prebieha nad už spracovanou sadou. Nakoľko je dátová sada veľmi nevyvážená, pri trénovaní sa vykonáva nadvzorkovanie.

6.2 ResNet50 s attention

Prvý experiment sa zameriava na vplyv použitia *self-attention* mechanizmu namiesto vrstvy združovania podľa priemeru. Boli natrénované dve neurónové siete, obe boli architektúry ResNet-50. Jedna sieť po konvolučných blokoch obsahovala vrstvu združovania podľa priemeru - *global average pooling*, za účelom agregovať výstupný tenzor pred vstupom do plne prepojenej vrstvy. V druhej sieti, vychádzajúc z návrhu v kapitole 4.2, bola táto vrstva nahradená *self-attention* mechanizmom. Implementácia mechanizmu je bližšie opísaná v kapitole 5.3. Obe siete boli natrénované na spracovanej dátovej sade určenej pre lokalizáciu dokumentov. Experimenty boli prevedené aj na pôvodnej, aj na novovzniknutej sade (výmena validačnej sady za časť trénovacej a naopak). Výsledky experimentov sa nachádzajú v tabuľkách 6.6 pre pôvodnú sadu a 6.7 pre novovzniknutú sadu. Z výsledkov je možné vyčítať, že model dosahuje lepšie výsledky na oboch validačných sadách za použitia *self-attention* mechanizmu na oboch úrovniach. Dá sa predpokladať, že váhované sčítanie vektorov, ako spôsob agregácie dovoluje, aby sa do výsledného agregovaného vektora dostali dôležité črty obrázka pre danú úlohu. Tabuľka 6.7 taktiež znázorňuje, že na novej validačnej sade, dosahujú siete výrazne lepšie výsledky. Môže to znamenať, že validačná sada vykazuje isté odlišnosti od tej trénovacej a je pravdepodobne náročnejšia, nakoľko pri trénovaní siete na pôvodných sadách sa sieť mala tendenciu pretrénovať, čo v prípade tejto modifikácie nenastalo. Spôsob agregácie výstupov pre výrezy je opísaný v implementačnej časti, tento spôsob bol realizovaný, nakoľko dosahoval najlepšie výsledky.

Trénovanie Obe siete boli trénované s veľkosťou trénovacej dávky 32 a učiacou konštantou nastavenou na počiatočnú hodnotu 0.0001. Táto konštanta sa počas trénovania postupne znižovala. Ako optimalizátor bol použitý optimalizátor Adam. Na výrezy bola počas trénovania aplikovaná dátová augmentácia, konkrétne afínne transformácie, zmena kontrastu, náhodný gausovský šum a zmena rozlíšenia výrezov. Bola zvolená miernejšia augmentácia obrázkov počas trénovania, pretože silné transformácie výrazne degradovali úspešnosť modelu.

[12] https://zenodo.org/record/784957628

Riešenie	Výrezy	Strany
ResNet-50 + GAP	62.3	75.6
ResNet-50 + attention	68.1	77.9

Tabulka 6.6: Výsledky experimentov sietí ResNet-50 s použitím *global average pooling* (GAP) a s použitím *self-attention* na validačnej sade na úrovni výrezov a úrovni celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na pôvodnej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosti - *accuracy*, v percentách.

Riešenie	Výrezy	Strany
ResNet-50 + GAP	79.8	86.2
ResNet-50 + attention	83.8	89.1

Tabulka 6.7: Výsledky experimentov sietí ResNet-50 s použitím *global average pooling* (GAP) a s použitím *self-attention* na validačnej sade na úrovni výrezov a úrovní celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na modifikovanej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosť - *accuracy*, v percentách.

6.3 Semi-supervizované učenie

Pseudo labeling Ďalší experiment sa venuje semi-supervizovanému experimentu s využitím celej dátovej sady. Spôsob anotácie obrázkov pomocou natrénovanej siete je opísaný v implementačnej časti práce (viď kapitolu 5). Daný experiment sleduje, či zvačšenie dátovej sady využívajúc metódy *pseudo labeling* dopomôže k zlepšeniu úspešnosti siete. Ako bolo spomenuté, po experimente s krížovou validáciou je zrejmé, že validačná sada je náročnejšia, predpokladalo sa teda, že rozšírenie dátovej sady o nové obrázky dopomôže k vylepšeniu na pôvodnej sade. Obe natrénované siete z predchádzajúceho experimentu sú použité k anotácii výrezov. Tieto isté siete sa následne znova natrénovali nad rozšírenou trénovacou sadou, pričom spôsob trénovania sietí sa nemenil. Výsledky týchto experimentov sa nachádzajú v tabuľkách 6.8 a 6.9. Z týchto výsledkov je zrejmé, že rozšírenie trénovacej dátovej sady použitím tzv. „pseudo anotácií“ prinieslo väčší dopad na sieť implementovanú s vrstvou združovania podľa priemeru, ako v prípade siete s mechanizmom *self-attention*, avšak v oboch prípadoch ide len o zanedbateľné zlepšenie. Dôvodom môže byť, že model na úrovni výrezov nedosahuje uspokojivé výsledky a spôsob vytvárania anotácii prebieha na tejto úrovni. Sieť sa teda učila aj predikovať triedy nesprávne anotovaných obrázkov pre pomerne veľkú časť rozšírenej sady. Z výsledkov je taktiež možné vyčítať, že rozdiely medzi výsledkami po vykonaní tohto experimentu sú zhruba rovnaké aj pre prípad modifikovanej dátovej sady.

Riešenie	Výrezy	Strany
ResNet-50 + GAP	63.6	76.9
ResNet-50 + attention	68.2	78.3

Tabulka 6.8: Výsledky semi-supervizovaného experimentou použitím metódy *pseudo labeling*, sietí ResNet-50 s použitím *global average pooling* (GAP) a s použitím *self-attention* na validačnej sade na úrovni výrezov a úrovní celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na pôvodnej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosť - *accuracy*, v percentách.

Riešenie	Výrezy	Strany
ResNet-50 + GAP	80.7	87.1
ResNet-50 + attention	84.1	89.3

Tabulka 6.9: Výsledky semi-supervizovaného experimentu použitím metódy *pseudo labeling*, sietí ResNet-50 s použitím *global average pooling* (GAP) a s použitím *self-attention* na validačnej sade na úrovni výrezov a úrovní celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na modifikovanej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosť - *accuracy*, v percentách.

Masked image modelling Nasledujúci experiment vychádza z použitia modelu BEiT pre predtrénovanie vizuálneho transformera na úlohu modelovanie maskovaných obrázkov, ktorý je následne dotrénovaný na danú klasifikačnú úlohu (viď 4.2). Sú natrénované dva modely. Jeden model je predtrénovaný modelovaním maskovaných obrázkov, len na podmnožine dátovej sady a druhý model je predtrénovaný na všetkých historických dokumentoch z danej dátovej sady. V oboch prípadoch sa ako tokenizér obrázkov použila natrénovaná sieť ResNet-50 s vrstvou združovania podľa priemeru pred plne prepojenou vrstvou, ktorej podvzorkovaný výstup sa nakvantizoval použitím zhlukovacej metódy *k-means*. Výsledky v tabuľkách 6.10 a 6.11 napovedajú tomu, že v prípade trénovania modelu len na podmnožine dátovej sady určenú k lokalizácii, dosiahol transformer podobné výsledky, ako konvolučná neurónová sieť. Na novej validačnej sade však model predtrénovaný na všetkých obrázkoch dátovej sady dosiahol lepšiu úspešnosť, ako konvolučná neurónová sieť s mechanizmom *self-attention*.

Obe siete boli trénované na klasifikačnú úlohu s veľkosťou trénovacej dávky 32 a učiacou konštantou nastavenou na počiatočnú hodnotu 0.0001. Táto konštanta sa počas trénovania postupne znižovala. Ako optimalizátor bol použitý optimalizátor Adam. Na výrezy boli počas trénovania aplikované rovnaké transformácie, ako pri konvolučných sieťach. BEiT model bol nakonfigurovaný nasledovne: bolo nastavených 12 skrytých vrstiev, parameter predstavujúci počet hláv *self-attention* mechanizmu bol taktiež nastavený na hodnotu 12, aktivačná funkcia bola nastavená na funkciu GeLU.

Dátová sada	Výrezy	Strany
Lokácia	62.7	76.1
Celá dátová sada	61.9	75.6

Tabulka 6.10: Výsledky semi-supervizovaného učenia použitím modelu BEiT na validačnej sade na úrovni výrezov a úrovní celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na pôvodnej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosť - *accuracy*, v percentách.

Dátová sada	Výrezy	Strany
Lokácia	81.3	87.8
Celá dátová sada	84.9	91.7

Tabulka 6.11: Výsledky semi-supervizovaného učenia použitím modelu BEiT, na validačnej sade, na úrovni výrezov a úrovní celých strán dokumentov. Tieto výsledky prislúchajú k experimentom konajúcimi sa na modifikovanej dátovej sade. Hodnoty v stĺpcoch predstavujú úspešnosti vyjadrené metrikou presnosť - *accuracy*, v percentách.

6.4 Zhrnutie výsledkov a testovanie

Výsledky všetkých prevedených experimentov z predchádzajúcej kapitoly nasvädčujú tomu, že validačná dátová sada dodaná organizátormi súťaže bola náročnejšia, resp. vykazovala známky odlišnosti od tej trénovacej. Naznačujú to výsledky experimentov vykonávajúcich krížovú validáciu naprieč sadám. Všetky natrénované siete dosahovali vyššiu úspešnosť na modifikovanej sade.

Ďalším poznatkom vyplývajúcim z experimentov je, že konvolučné neurónové siete dosahovali lepšie výsledky na pôvodnej dátovej sade, zatiaľ čo model BEiT dosiahol na modifikovanej sade výraznejšie zlepšenie. Osvedčila sa aj stratégia predspracovania dátovej sady na výrezy v tvare obdĺžnika. Klasická konvolučná neurónová sieť, dosahuje na validačnej sade podobné úspešnosti, ako v prípade systémov PERO, ktoré kombinovali model nad textovými riadkami s modelom nad výrezmi fixnej veľkosti, na konferencii ICDAR 2021 (viď kapitola 2.5).

Posledná fáza zahŕňa evaluáciu najúspešnejšieho modelu na úrovni strán dokumentu na testovacej sade poskytnutej organizátormi súťaže. Nakoľko sa experimenty vykonávali na dvoch sadách, vyhodnotené boli dva modely. Prvý model, konvolučná neurónová sieť architektúry ResNet-50 natrénovaná na rozšírenej dátovej sade využitím metódy *pseudo labeling*, dosahoval najlepšie výsledky na pôvodnej dátovej sade. Model, ktorý dosiahol najlepší výsledok na modifikovanej dátovej sade bol BEiT, predtrénovaný na všetkých obrázkoch z dátovej sady. Z tohto je možné odvodiť, že semi-supervizované učenie dopomohlo k miernemu vylepšeniu modelov v tejto úlohe.

Konvolučná neurónová sieť natrénovaná nad rozšírenou trénovacou dátovou sadou dosiahla na testovacej sade presnosť 81.6% a BEiT predtrénovaný na celej dátovej sade dosiahol presnosť 82.9%. Nakoľko spomínaný víťazný systém PERO dosiahol na testovacej sade presnosť 79.4%, v oboch prípadoch ide o mierne zlepšenie.

Kapitola 7

Záver

Cieľ tejto práce je vytvoriť systém postavený na hlbokých neurónových sieťach, na klasifikáciu historických dokumentov. Práca sa sústreďuje na niekoľko úloh: klasifikácia podľa typu písaného písma, klasifikácia podľa typu tlačeného písma, datovanie dokumentov do obdobia ich vzniku a klasifikácia dokumentov podľa miesta vzniku. Z analýzy existujúcich metód, ktoré vychádzajú z konajúcich sa súťaží vyplynulo, že klasifikácia dokumentov podľa miesta vzniku je najnáročnejšou úlohou. Z tohto dôvodu sa navrhnuté riešenia týkajú práve tejto úlohy.

Navrhnuté systémy pracujú nad výrezmi v tvare obdĺžnika, ktoré sú extrahované z pôvodnych obrázkov. Výstup navrhnutých sietí teda predstavuje triedu jedného výrezu, tieto výstupy sú následne agregované do jednotnej predikcie.

Hlavnou súčasťou tejto práce je semi-supervizované učenie. Z dôvodu, dostupnosti mnohých dátových sád historických dokumentov sa tieto neanotované dokumenty použili pri realizácii niekoľkých navrhnutých systémov. Prvý navrhnutý systém je postavený na modeli BEiT. Tento model bol použitý k predtrénovaniu vizuálneho transformera na úlohu modelovanie maskovaných obrázkov a bol následne dotrénovaný na klasifikačnú úlohu. Ďalší navrhnutý systém je postavený na hlbokej konvolučnej neurónovej sieti architektúry ResNet-50. V rámci tejto siete bola navrhnutá modifikácia predposlednej združovacej vrstvy. Namiesto vrstvy združovania podľa priemeru je realizovaný mechanizmus *self-attention*, ktorý agreguje výstup siete pred plne prepojenou vrstvou.

V rámci analýzy bolo nájdených niekoľko dátových sád, ktoré boli zverejnené organizátormi súťaží. Na vykonávanie experimentov sa použila práve dátová sada z konferencie ICDAR 2021. Táto dátová sada obsahuje štyri časti pre spomenuté štyri úlohy. V rámci experimentov bola modifikovaná dátová sada pre úlohu lokalizovania dokumentov. Po prevedenej krížovej validácii z výsledkov experimentov vyplynulo, že poskytnutá validačná sada organizátormi súťaže, je výrazne odlišná od trénovacej. Z tohto dôvodu sa každý experiment vykonával na dvoch sadách, na tej pôvodnej a na modifikovanej.

Z výsledkov experimentov vyplynulo, že realizácia *self-attention* mechanizmu dopomohla k zlepšeniu úspešnosti modelu. Toto naznačuje, že do klasifikačnej vrstvy sa týmto spôsobom dostávajú relevantnejšie informácie z obrázka, ako pri klasickom priemerovaní vektorov. Konvolučné neurónové siete boli použité aj k semi-supervizovaným experimentom prostredníctvom metódy *pseudo labeling*, kedy sa natrénované siete použili k vytváraniu anotácií. Siete sa následne znova natrénovali, ale už na rozšírenej dátovej sade. Tento systém dosiahol najlepšiu úspešnosť na pôvodnej dátovej sade, pričom systém BEiT natrénovaný na celej dátovej sade, dosiahol najlepšiu úspešnosť práva na tej modifikovanej. Tieto dva systémy boli následne evaluované na poskytnutej testovacej sade a porovnané s víťazným systémom

PERO, založeným na fúzii dvoch systémov. Pri oboch systémoch nastalo mierne zlepšenie oproti systému PERO, nakoľko systém postavený na modelovaní maskovaných obrázkov dosiahol presnosť 82.9 %, systém postavený na konvolučnej neurónovej sieti s mechanizmom *self-attention* dosiahol 81.6 % a presnosť spomínaného víťazného systému je 79.4 %. V tomto prípade je teda zlepšenie o 3.5 %.

V rámci ďalšej práce by bolo vhodné vyskúšať v konvolučnej neurónovej sieti mechanizmus *multihead self-attention*, teda *self-attention* s viacerými hlavami, výstupy pre každú hlavu by sa následne agregovali. Ďalej by sa taktiež mohla použiť *self-attention* namiesto vrstvy združovania podľa priemeru v modeli BEiT trénovaného na klasifikáciu.

Literatura

[1] Arazo, E., Ortego, D., Albert, P., O'Connor, N. E. a McGuinness, K. Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. 2020, s. 1–8. DOI: 10.1109/IJCNN48605.2020.9207304.

[2] Bao, H., Dong, L., Piao, S. a Wei, F. *BEiT: BERT Pre-Training of Image Transformers*. 2022.

[3] Cloppet, F., Eglin, V., Helias Baron, M., Kieu, C., Vincent, N. et al. ICDAR2017 Competition on the Classification of Medieval Handwritings in Latin Script. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 2017, sv. 01, s. 1371–1376. DOI: 10.1109/ICDAR.2017.224.

[4] Cloppet, F., Églin, V., Kieu, V. C., Stutzmann, D. a Vincent, N. ICFHR2016 Competition on the Classification of Medieval Handwritings in Latin Script. In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. 2016, s. 590–595. DOI: 10.1109/ICFHR.2016.0113.

[5] Constantopoulos, P., Doerr, M., Theodoridou, M. a Tzobanakis, M. Historical documents as monuments and as sources. In: *Computer Applications and Quantitative Methods in Archaeology Conference (CAA 2002), Heraklion, April 2002*. 2002.

[6] Devlin, J., Chang, M.-W., Lee, K. a Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, červen 2019, s. 4171–4186. DOI: 10.18653/v1/N19-1423. Dostupné z: https://aclanthology.org/N19-1423.

[7] Devlin, J., Chang, M.-W., Lee, K. a Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019.

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X. et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021.

[9] Dumoulin, V. a Visin, F. A guide to convolution arithmetic for deep learning. 2018.

[10] He, K., Zhang, X., Ren, S. a Sun, J. *Deep Residual Learning for Image Recognition*. 2015.

[11] Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R. et al. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. IEEE. 2021, sv. 29, s. 3451–3460.

[12] Imambi, S., Prakash, K. B. a Kanagachidambaresan, G. PyTorch. *Programming with TensorFlow: Solution for Edge Computing Applications*. Springer. 2021, s. 87–104.

[13] Kestemont, M., Christlein, V. a Stutzmann, D. Artificial paleography: computational approaches to identifying script types in medieval manuscripts. *Speculum*. University of Chicago Press Chicago, IL. 2017, sv. 92, S1, s. S86–S109.

[14] Kodym, O. a Hradiš, M. *Page Layout Analysis System for Unconstrained Historic Documents*. 2021.

[15] Kölsch, A., Afzal, M. Z., Ebbecke, M. a Liwicki, M. Real-Time Document Image Classification Using Deep CNN and Extreme Learning Machines. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. 2017, sv. 01, s. 1318–1323. DOI: 10.1109/ICDAR.2017.217.

[16] Liu, L., Jiang, H., He, P., Chen, W., Liu, X. et al. On the Variance of the Adaptive Learning Rate and Beyond. In: *International Conference on Learning Representations*. 2020. Dostupné z: https://openreview.net/forum?id=rkgz2aEKDr.

[17] Martin Kišš, K. B. a Hradiš, M. Importance of Textlines in Historical Document Classification. 2021.

[18] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C. et al. *Zero-Shot Text-to-Image Generation*. 2021.

[19] Ridha Ilyas, B., Beladgham, M., Merit, K. a ahmed, A. taleb. Illumination-robust face recognition based on deep convolutional neural networks architectures. Prosinec 2019, Vol 18, s. 1015 1027. DOI: 10.11591/ijeecs.v18.i2.pp1015-1027.

[20] Sengupta, A., Ye, Y., Wang, R., Liu, C. a Roy, K. Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in neuroscience*. Frontiers Media SA. 2019, sv. 13, s. 95.

[21] Seuret, M., Limbach, S., Weichselbaumer, N., Maier, A. a Christlein, V. Dataset of pages from early printed books with multiple font groups. In: *Proceedings of the 5th international workshop on historical document imaging and processing*. 2019, s. 1–6.

[22] Seuret, M., Limbach, S., Weichselbaumer, N., Maier, A. a Christlein, V. Dataset of pages from early printed books with multiple font groups. In: *Proceedings of the 5th international workshop on historical document imaging and processing*. 2019, s. 1–6.

[23] Seuret, M., Limbach, S., Weichselbaumer, N., Maier, A. a Christlein, V. Dataset of Pages from Early Printed Books with Multiple Font Groups.

In: *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*. 2019, s. 1–6.

[24] Seuret, M., Nicolaou, A., Rodríguez Salas, D., Weichselbaumer, N., Stutzmann, D. et al. ICDAR 2021 Competition on Historical Document Classification. In: Lladós, J., Lopresti, D. a Uchida, S., ed. *Document Analysis and Recognition – ICDAR 2021*. Cham: Springer International Publishing, 2021, s. 618–634. ISBN 978-3-030-86337-1.

[25] Simonyan, K. a Zisserman, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015.

[26] Simonyan, K. a Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *International Conference on Learning Representations*. 2015.

[27] Targ, S., Almeida, D. a Lyman, K. *Resnet in Resnet: Generalizing Residual Architectures*. 2016.

[28] Thorpe, M. a Gennip, Y. van. *Deep Limits of Residual Neural Networks*. 2020.

[29] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention is All You Need. In:. 2017. Dostupné z: https://arxiv.org/pdf/1706.03762.pdf.

[30] Vedaldi, A. a Zisserman, A. Vgg convolutional neural networks practical. *Department of Engineering Science, University of Oxford*. 2016, sv. 66.

[31] Zhuang, B., Liu, J., Pan, Z., He, H., Weng, Y. et al. *A Survey on Efficient Training of Transformers*. 2023.

Přílohy

Příloha A

Obsah DVD

Obsah priloženého DVD:

- `source_code/` - priečinok obsahujúci zdrojové kódy
  - `BEiT/` - skripty k systému BEiT
  - `CNN/` - skripty ku konvolučným neurónovým sieťam
  - `evaluation/` - skript k evaluácii
  - `image_processing/` - skript ku spracovaniu dátovej sady
  - `requirements.txt` - súbor s požadovanými knižnicami
- `text/` - priečinok obsahujúci text práce
- `video/` - priečinok obsahujúci video