**Mojou** **temou** je Spracovanie medicínskych dát umelou inteligenciou pre podporu lekárskej diagnózy, pricom neskor sme sa rozhodli, ze budeme porovnavat rozne principy tvorenia filtrov v prvych vrstvach konvolucnych neuronovych sieti. Ako manualne zadavanie tychto filtrov, sme vybrali gaborove filtre a pre automaticke vytvaranie filtrov, sme vybrali backpropagation, autoencoder a transfer learning.

**Mojou domenou** boli medicinske data, konkretne som si vybral histologicke data, pricom sme vyuzili dataset PCAM, teda PatchCamelyon, ktory obsahuje …trenovacich, testovacich a validacnych fotiek histologickych dat, pricom su to rozne organy, tieto obrazky su v tvare 96x96x3. Ukazku tychto dat mozme vidiet na obrazku c.1. Neuronova siet ktoru sme vytvorili, aj napriek celkom dobrym vysledkom sa nedokazala z dat naucit dostatok informacii, a preto sme boli nutení vybrat si dataset ktory bude obsahovat jednoduchsie obrazky aby sme dokazali objektivne porovnat jednotlive principy. Pre tuto skutocnost sme nasli dataset DAGM, ktory obsahuje len ciernobiele obrazky v tvare 512x512. Tento dataset obsahuje 10 tried, ktore sa od seba lisia. My sme si zvolili len dve a to konkretne tie ktore su na obrazku c.2 a c.3. Pricom tento dataset bol prilis velky a oproti povodnemu dataset PCAM, sme ho museli upravit a to tak ze sme ho nasekali na velkost 96x96, pricom sme presuvali o 16 bitov, a tak sme vytvorili z povodnych 575 obrazkov na 56108, pricom takyto pocet mame aj pre validacne a testovacie data (dokkopy)

**Pribuzne riesenie** k mojmu je Rotation Equivariant CNNs for Digital Pathology, ktore sa nachadza na githube PCAM datasetu, pricom toto riesenie nepouziva klasicku konvolucnu siet ale upravenu s nazvom G-CNN, pricom tato konvolucna siet je generalizaciou klasickej, ktoré sú ekvivalentne v rámci všeobecnejších skupín symetrie. ako je skupina p4 ktora zahrna 90 stupnovu rotaciu.

Pre p4 a p4m to jednoducho znamená ze kanály prichádzajú do skupín po 4 alebo 8, čo zodpovedá 4 čistým rotáciám v p4

alebo 8 roto-odrazov v p4m.

Nasledne pomocou tychto vrstiev G-CNN vytvaraju pomocou Dense vrstvie takzvanu DenseNet, model tejto architektury mozme videl na obrazku c 4.

Architektura je zalozena na hustoprepojenych konvolucnych sieti, ktoré používajú zásobník všetkých predchádzajúcich vrstiev ako vstup striedaný s prechodovými blokmi pozostávajúca z 1 × 1 konvolučnej vrstvy a 2 × 2 pruhovaného priemerného združovania (avaragepooling).

**Ako vlastne riesenie** sme otestovali rozne zakladen architektury ako VGG16 (50), RESNET (velkost 8 67,2) a UNET (velkost 9 80), taktiez sme vyskusali aj jednoduchsie NN ktore obsahovali maximalne 5 konvolucnych vrstiev, no tie nepreukazali nejake rozdielne zmeny a maximalne dosiahli  74,9% uspesnost, pricom jednotlive variacie tychto sieti mozete vidiet v tabulke c.1. Najlepsie z tychto architektur a celkovo modelov ktore sme vyskusali bolo UNET s 9 konv vrstvami. Nasledne tuto architekturu sme pouzili ako zaklad pre dalsie porovnavania s tym ze zakazdym vlozime jednu vrstvu pre unet. A tato jedno vrstva tvori filtre nejak specialne, teda podla spominanych principov.

Transfer learning, pri tomto principe sme pouzili rovnaky typ architektury a to tak ze sme pouzili uz vytvorene siet a uz aj s naucenymi vahami, ktoru sme nasledne vlozili pred UNET architekturu. V tejto sieti sme definovali pre prvych 15 vrstiev ze sa nemozu upravovat filtre. Jednotlive vysledky mozte vidiet tu

Pri dalsich rieseniach sme uz robili rozsiahlejsie porovnavanie a to take ze sme pre kazdy princip trenovali siet ktora obsahovala rozne velkosti filtrov a rozny pocet filtrov.

**Backpropagation**, tento princip je implementovany tak, ze sme ako prvu vrstvu pouzili klasicku konvolucnu vrstvu pre UNET siet. Teda Conv2D, Batch Normalization, RELU Activation funkcia, s druhou skrytou vrstvou zloženou z Conv2D, Batch Normalization, RELU Activation funkciou. Za túto vrstvu sme vlozili MaxPooling2D s velkostou 2 a dropout vrstvu pre znizenie preucenia.

Pre tento princip sme vybrali velkost filtrov 3, 5 a 7 a pocet filtrov 16, 32 a 64.

Pre PCAM vidime vysledky v tabulke 2. a pre DAGM vidime vysledky v tabulke 3.

**Autoencoder**, tento pricnip je implementovany tak, Conv2D, Dropout, Conv2D, MaxPooling2D vo velkosti 2 aConv2D, Dropout, Conv2D, s poslednou vrstvou ktora zlucuje pomocou Concatenate a UpSampling2D s velkostou 2.

Pre tento princip sme vybrali velkost filtrov 3, 5 a 7 a pocet filtrov 16, 32 a 64.

Pre PCAM vidime vysledky v tabulke 4. a pre DAGM vidime vysledky v tabulke 5

**Gaborove** filtre, tento pricnip je implementovany tak, ze parameter ktory hovori o moznosti naucenia novych fitlrov je nastaveny na False, nasledované vytvorime Gaborove filtre pomocou knižnice OpenCV, a tieto filtre neskor ulozime ako vahy danej konvolucnej vrstvy

Pre tento princip sme vybrali velkost filtrov 5, 7 a 9 a pocet filtrov 16, 32 a 64.

Pre PCAM vidime vysledky v tabulke 6. a pre DAGM vidime vysledky v tabulke 7

Autoencoder na testovacich datach dosiahol vysledky pri roznych pametrov podobne vysledke ako vidime v tabulke presnost (accurancy) je skoro vo vsetkych pripadov 95, pricom jednotlive parametre maju rozdielnu presnost len v desatinych cislam. Okrem zlozenie parametrov velkost filtrov 5x5 a pocet filtrov 32 tato kombinacia dosiahla najmensiu presnost a to 92,5 %. Pricom najlepsie vysledky dosiahla kombinacia parametrov 7x7 velkost filtrov a 32 pocet filtrov a to 95,88%.

Backpropagation v tomto pripade sa jednotlive vysledky uz lisia nie len v desatinnych ale aj v jednotkach najlepsie vysledky boli pre kombinaciu 7x7 a 64 s presnostou 95,2 a najhorsia kombinacia bola 3x3 a 16 s vysledkom 91,49% presnost.

Gaborove filtre dosahovali podobne vysledky pricom vacsina z tych najlepsich sa lisi len v desatinnych miestach ale najhorsie sa zhorsili aj o jednotky percent pricom najhorsia kombinacia je 7x7 a 32 s uspesnostou 91,8 a najlepsia je kombinacia 5x5 a 64 s percentami presnosti 93,55%.

pricom vsetky vysledky su len brane z dat ktore siet pri uceni nevidela a su rozdele v pomere 2:1 (negativne:pozitivne) okolo 21 tisic obrazkov. A ako mozme vyhodnotit vsetky principe maju velmi dobru uspesnost minimalny pocet tried zle pozaradovali, pricom najlepsi princip je Autoencoder, Pricom ked toto budeme porovnavat aj s Transfer learningom stale ostava autoencoder ako najlepsi princip.