

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Návrh metody strojového vidění pro třídění mincí různých měn

Diplomová práce

Vedoucí práce:
Ing. Jan Kolomazník, Ph.D.

Bc. Jana Čupová

Brno 2016

zadani prace

Text poděkování Ráda bych poděkovala vedoucímu práce Ing. Janu Kolomazníkovi Ph.D. a konzultantovi Ing. Richardu Kleinovi za odborné vedení, cenné rady a připomínky k vypracování této diplomové práce. Také děkuji jejich kolegům za odborné konzultace, přátelům a rodině za podporu.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Návrh metody strojového vidění pro třídění mincí různých měn**

vypracovala samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědoma, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 16. května 2016

.....

Abstract

Čupová, J. Design methods machine vision for sorting coins of different currencies. Diploma thesis. Brno: Mendel University, 2016.

This thesis is aiming to design method that would be suitable for coin recognition by given picture. There are implemented and tested several methods used for computer vision, all of them are written in C++ language. Hardware part, model, used for sorting coins is composed by plastic and additional electronics components. There is implemented application to count amount of sorted coins. It is done in development environment Control Web with extension VisionLab. This application is connected with model through Datalab, that is industrial system input and output. Theoretical part contains description of methods used for image recognition. Following practical part of this thesis is focused on description of implementation each method and test results of image recognition of coins.

Keywords computer vision, SIFT, SURF, artificial neural network, hopfield network, multilayer neural network, control web, datalab, c++

Abstrakt

Čupová, J. Návrh metody strojového vidění pro třídění mincí různých měn. Diplomová práce. Brno: Mendelova univerzita, 2016.

Diplomová práce si klade za cíl navrhnout způsob, který by byl vhodný pro rozpoznávání mincí podle daného obrázku. Jsou zde implementovány a otestovány metody pro počítačové vidění v jazyce C++. Hardwarová část práce popisuje model. Slouží k třídění mincí složen z plastových dílů a elektronických prvků. Je zde implementována aplikace na počítání mincí, vytvořena ve vývojovém prostředí Control Web s rozšířením VisionLab. Aplikace s modelem je propojena jednotkou průmyslového systému vstupních a výstupních modulů Datalab. V teoretické části se nachází popis metod rozpoznávání obrazu. V praktické části je popsána implementace těchto metod a jejich výsledky při rozpoznávání mincí.

Klíčová slova SIFT, SURF, umělá neuronová síť, hopfieldova síť, vícevrstvá neuronová síť, control web, datalab, c++

Obsah

1	Úvod	8
1.1	Cíle diplomové práce	8
2	Teoretická část	10
2.1	Pohon otáčecího ramene	10
2.1.1	Krokový motor	10
2.1.2	Servomotor	11
2.1.3	Ovladač krokového motoru DRV8825	12
2.2	Zpracování obrazu	13
2.2.1	Prahování	13
2.2.2	Cannyho hranová detekce	15
2.3	Metody rozpoznávání obrazu	16
2.3.1	FLANN - Knihovna přibližného rychlého vyhledávání pro porovnání nejbližších sousedů	16
2.3.2	FAST - Features from accelerated segment test	16
2.3.3	SIFT - Scale-invariant feature transform	18
2.3.4	SURF - Speeded-Up Robust Features	22
2.4	Umělé neuronové sítě	26
2.4.1	Neuron	27
2.4.2	Struktura neuronové sítě	27
2.4.3	Učení neuronové sítě	28
2.5	Hopfieldova síť	29
2.5.1	Uspořádání Hopfieldovy sítě	30
2.5.2	Učení a vybavování Hopfieldovy sítě	30
2.5.3	Hebbovo adaptační pravidlo	31
2.5.4	Vlastnosti Hopfieldovy sítě	32
2.6	Vícevrstvá neuronová síť	32
2.6.1	Učení vícevrstvé neuronové sítě	34
2.6.2	Backpropagation	34
3	Metodika	35
3.1	Visual Studio	35
3.1.1	OpenCV	35
3.2	Control web	35
3.3	Matlab	36
3.4	Snímací kamera	36
3.5	Datalab IO	37
4	Praktická část	38
4.1	Seznam rozpoznávaných měn	38
4.2	Model pro třídění mincí	39
4.3	Napájení modelu	40

4.4	Elektromagnety a brány pro průchod mincí	40
4.5	Krokový motor SX16-0402N	40
4.6	Řízení krokového motoru	41
4.7	Spínače pro detekci zastavení motoru	43
4.8	Osvětlení	43
4.9	Konektor	44
4.10	Uspořádání pořadí mincí	45
4.11	Rozpoznání mincí v aplikaci	45
4.12	Aplikace v Control web	46
4.13	Výpočet počtu kroků motoru	47
4.14	Moduly pro ovládání prvků	47
4.15	Rozpoznávání obrazu	48
4.16	Předzpracování obrázu pro metody SURF a SIFT	48
4.17	FLANN knihovna s metodou SURF	48
4.18	FLANN knihovna s metodou SIFT	51
4.18.1	Závěr k metodě SIFT	53
4.19	Hopfieldova síť	54
4.19.1	Vstupní data	54
4.19.2	Předzpracování obrazu pro Hopfieldovou síť	54
4.19.3	Trénovací množina	55
4.19.4	Rozhodování	55
4.19.5	Natočení vstupních dat pro rozhodování	56
4.19.6	Určení hodnoty mince	57
4.19.7	Závěr Hopfieldovy sítě	59
4.20	Vícevrstvá neuronová síť	59
4.20.1	Načtení vstupních obrazů	59
4.20.2	Učení neuronové sítě pomocí Matlabu	61
4.20.3	Načtení vah do proměnných	64
4.20.4	Rozhodování vícevrstvé neuronové sítě	64
4.20.5	Výsledky vícevrstvé neuronové sítě	65
4.20.6	Závěr vícevrstvé neuronové sítě	67
5	Závěr	68
6	Reference	70
7	Příloha	76
7.1	Seznam mincí	76
7.2	Přehled obrázků k praktické části	81

1 Úvod

Strojové vidění je v nynější době nepostradatelné a využívá se v mnoha odvětvích. Například v automatizaci, kde se například kontroluje úplnost balení, třeba u léků, dále u automatické kontroly sestavení a montáže různých dílů, identifikace barev v potravinářství, rozpoznání a polohování, zda je před montováním správně umístěno například víčko na PET láhvi, měření a kontrole různých tolerancí. V automobilovém průmyslu se pomocí čtení a verifikace kódů rozpoznávají různé automobilové díly a také se pomocí strojového vidění kontroluje kvalita povrchu a potisků. Například poškozování hran, poškrábání povrchů, vady nátěrů či nedokonalý tisk. Dále se se strojovým viděním můžeme setkat u rozpoznávání SPZ na automobilech. Sektor bankovníctví zatím strojové vidění pro rozpoznávání mincí nepoužívá. Jeden ze způsobů, které se nyní používají pro rozpoznávání mincí, je dle váhy, kde je však problém při mechanickém poškozování mince, kdy dojde ke změně její váhy. Druhým způsobem rozpoznávání mincí jsou mechanické třídičky, které jsou založeny na průměru mince. Tyto třídičky jsou však finančně náročnější a také objemné a těžké. V mé dřívější bakalářské práci jsem mince třídila pomocí průměru, ale muselo jít vždy o jednu měnu třídění. V případě různých měn se může stát, že různé mince mají stejný průměr a nelze je tímto způsobem od sebe rozpoznat.

V minulosti byly vyzkoušeny různé způsoby rozpoznávání pomocí senzoru. Někteří se snažili rozpoznávat mince také pomocí rozměru, ale došli ke stejnému závěru jako v mé bakalářské práci. Jejich model byl v tomto rozpoznávání o hodně pomalejší, jelikož do modelu třídičky byla vždy vpuštěna jen jedna mince a druhá mohla být vpuštěna až po konečném umístění předchozí mince. Aplikační část rychlost rozpoznávání neovlivnila. Dále se jeden student ve své bakalářské a diplomové práci pokoušel o rozpoznávání mincí přímo pomocí obrazu, kdy ale jen porovnával obrazy a neměl k této práci žádný model pro třídičku mincí. Aplikace byla naprogramovaná v jazyce PHP s databází SQL, kde měl uložené všechny potřebné mince s jejich 120 různými natočeními. Tímto se tedy dostal k problému využití velkého množství paměti a také pomalému porovnávání, jelikož pro každou minci procházel 120 obrazy.

1.1 Cíle diplomové práce

Cílem této diplomové práce je upravit model na třídění mincí z bakalářské práce a uzpůsobit jej pro rozpoznávání mincí pomocí obrazu. Pro tyto úpravy také změnit aplikaci vytvořenou v Control webu. Dále bude tato práce mít za cíl umět rozpoznávat mince všech měn, které Česká národní banka nabízí k výměně. Množství těchto roztríděných mincí se bude ukládat do archivního souboru pro pozdější možné kontroly, ale také tyto počty uživatel uvidí hned v aplikaci. Uživatel si bude moci v aplikaci vybrat, kterou z měn chce aktuálně třídít či případně vybrat několik měn, které bude třídít mezi sebou. Základem rozpoznávání mincí mezi sebou bude jejich rozměr. Pro případ, že by uživatel vložil do modelu pro třídění něco jiného než minci, se bude ještě rozpoznávat mince pomocí obrazu. Detekce podvodů není cílem této

práce. V případě, že některé mince budou mít stejný rozměr, jako například 10 Kč a 1 Euro, budou rozpoznávány od sebe obrazem. Bude tedy nutné vybrat několik metod, které by byly vhodné pro vyzkoušení rozpoznávání mincí pomocí obrazu. Metoda s nejlepšími výsledky bude pak vybrána jako vhodná pro třídění mincí. Obraz bude snímán FullHD webovou kamerou připevněnou k třídícímu modelu.

Pro naplnění cíle práce je nutné splnit následující body:

1. zamyslet se nad úpravou modelu pro třídění mincí
2. upravit aplikaci pro třídění mincí vytvořenou v programu Control web, aby komunikovala s upraveným modelem
3. analyzovat metody pro rozpoznávání obrazu
4. vybrat a implementovat vhodné metody pro rozpoznávání obrazu
5. otestovat metody pro rozpoznávání obrazu a navrhnout nejlepší

2 Teoretická část

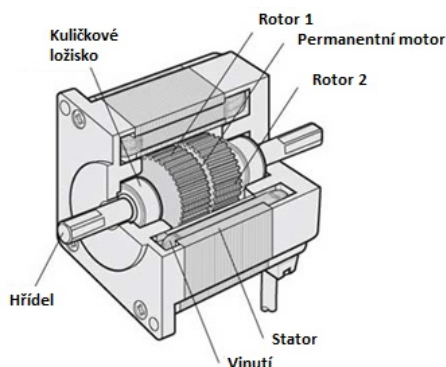
V této kapitole jsou popsány teoretické znalosti jak k tvorbě hardwarové části, tak k tvorbě softwarové části. Pro hardwarovou část se tu například dozvíme o krokovém motoru a servomotoru. Je zde i část o zpracování obrazu, kde je znázorněno prahování a Cannyho detekce hran. Největší část této kapitoly je věnována rozpoznávání obrazu. Jsou zde popsány metody FLANN - Knihovna přibližného rychlého vyhledávání pro porovnání nejbližších sousedů, FAST - Features from accelerated segment test, SIFT - Scale-Invariant feature transform, SURF - Speed-Up Robust Features. Další podčástí této kapitoly jsou umělé neuronové sítě, kde jsou popsány Vícevrstvé neuronové sítě a Hopfieldova síť.

2.1 Pohon otáčecího ramene

V části této práce, která se zabývá hardwarem, bylo potřeba připojit k pohonu třídícího ramene motorek. K tomu se daly využít dvě možné varianty, a to krokový motor a servomotor.

2.1.1 Krokový motor

Krokové motory jsou motory pohybující se po jednotlivých krocích, protože jsou napájeny impulsně. Frekvence jednotlivých kroků je dána rychlostí impulsů, a od toho se také odvíjí rychlost otáčení motoru. Krokovým motorem se přesně určuje nastavování polohy a také rychlosti. Vinutí u krokového motoru je umístěno na statoru a motor nemá žádné kluzné kontakty. Komutace, bez které by nevzniklo točivé magnetické pole, se provádí v elektronickém ovladači. Ten ale není součástí krokového motoru. Jednoduché ovladače krokových motorů jsou konstruovány spínacími polovodičovými součástkami. Ty složitější využívají specializované obvody z polovodičových součástek. K řízení krokového motoru se využívá také ovladač řízený jednočipovým mikroprocesorem nebo výstupy z PLC systémů. (Uhlíř, 2007).



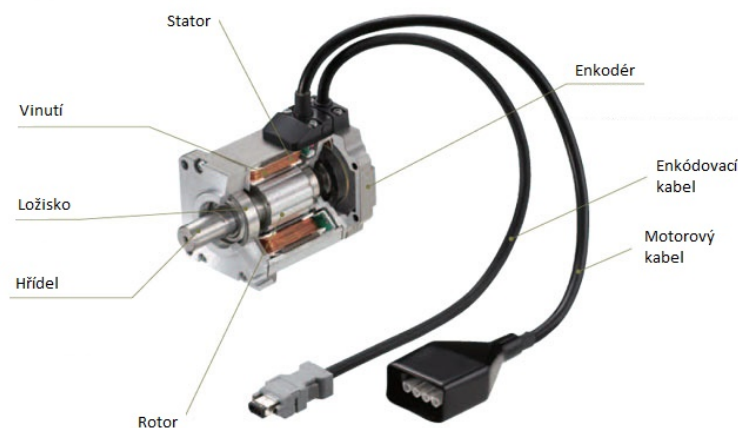
Obrázek 1: Krokový motor (Stepper Motor Overview, 2016)

2.1.2 Servomotor

Servomotor je motor pro elektrické, ale také hydraulické a pneumatické pohony, u kterých lze nastavit přesnou polohu natočení osy. Pomocí servomotoru lze ovládat například posuvy CNC strojů či nastavení čtecí hlavy u pevného disku (Vavříňák, 2006).

Servomotor se skládá z několika částí (Servo Motor Overview, 2016):

- Stator - rotující magnetické pole je vytvořeno pro účinné generování točivého momentu
- Vinutí - proud teče ve vinutí a vytváří se točivé magnetické pole
- Ložisko - kuličkové ložisko
- Hřídel - přenáší výstupní výkon motoru
- Rotor - permanentní magnet je umístěn externě na hřídel
- Enkodér - optický snímač, sleduje počet otáček a polohu hřídele
- Enkodovací kabel
- Motorový kabel



Obrázek 2: Servomotor (Servo Motor Overview, 2016)

Enkodér je sensor pro detekování rychlosti a pohybu motoru. Světlo z LED diody prochází detekcí polohy vzoru na šterbinovém disku a čte prvek pro přijímání světla. Vzory jsou pro absolutní snímání polohy a jsou závislé na úhlu otáčení enkodéru. Aktuální data o poloze se přenáší do ovladače servomotoru přes sériový přenos. (Servo Motor Overview, 2016)

Elektrické servomotory

Tyto servomotory jsou řízeny hlavně tranzistorovými měniči s pulzní šířkovou modulací. Pomocí fotoelektrického snímače je možné zajistit polohu hřídele servomotoru. Signál snímače polohy se zpětně přivede na regulátor a porovná se skutečná poloha motoru s požadovanou polohou. Na základě rozdílu těchto dvou poloh regulátor řídí měnič a nastavuje motor do požadované polohy. (Vavříňák, 2006)

Segmentové synchronní servomotory

Segmentové servomotory vycházejí z provedení statoru servomotoru. Jednotlivé fáze se navinou na segmenty statoru, z kterých se vytvoří svazek. Segmentové složení statorového svazku umožňuje navíjet každý segment vinutí zvlášť. Navíjení segmentů umožňuje dosažení lepšího plnění vinutí, a tím zhuštění magnetického pole, které je generované statorem. (Vavříňák, 2006)

2.1.3 Ovladač krokového motoru DRV8825

Jedna z možností řízení krokového motoru je pomocí ovladače DRV8825. Ten lze použít nejen pro řízení bipolárních krokových motorů, ale také k řízení krokových motorů do tiskáren, skenerů a dalších zařízení v automatizaci.

Tento ovladač vyžaduje napětí motoru v rozpětí 8,2 V do 45 V, které musí být připojeno přes VMOT a GND. Aby byl dodáván očekávaný proud pro krokový motor, je potřeba zapojení blokovacího kondenzátoru. Použitím tohoto kondenzátoru se zamezí možnosti trvalého poškození ovladače, jelikož může hodnota jmenovitého napětí překročit 45 V, i když napájecí napětí motoru je jen 12 V. (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

Čtyř-, šesti a osmidrátové krokové motory mohou být poháněny ovladačem DRV8825 v případě správného zapojení, které je nutné ověřit si v katalogu krokového motoru. Při připojení či odpojení krokového motoru v době, kdy je ovladač napájen, může dojít ke zničení ovladače.

Krokové motory mají specifikovanou velikost kroku, a to buď ve stupních na jeden krok anebo počtem kroků na jednu celou otáčku (360°). Ovladač DRV8825 umožňuje tyto kroky ještě zmenšit pomocí mezikroků. Například řízení motoru s 200 kroky na otáčku lze rozdělit na 800 mikrokroků na otáčku pomocí 4 různých úrovní. Nastavování úrovní se provádí nastavováním PINů MODE1, MODE2 a MODE3, a lze vybírat ze šesti různých kroků. (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

Každý impuls, který je na vstupu, odpovídá jednomu mikrokroku, který je nastaven pomocí pinů MODE. V případě, že piny MODE nejsou využity, je jeden impuls roven jednomu celému kroku. Tyto impulsy se dávají na vstup pinu s označením STEP. Směr otáčení krokového motoru se určuje na vstupu DIR. V případě, že potřebujeme, aby se otáčel v protisměru, nastavíme na vstup hodnotu 0. Pokud chceme krokový motor řídit jen v jednom směru, můžeme tento vstup úplně odpojit. (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

MODE0	MODE1	MODE2	KROK
low	low	low	celý krok
high	low	low	poloviční krok
low	high	low	1/4 kroku
high	high	low	1/8 kroku
low	low	high	1/16 kroku
high	low	high	1/32 kroku
low	high	low	1/32 kroku
high	high	high	1/32 kroku

Tabulka 1: Nastavení velikosti kroku krokového motoru (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

2.2 Zpracování obrazu

Dříve než dojde k samotnému rozpoznávání obrazu, je nutno tento upravit. V této práci není potřeba porovnávat barevné obrazy, a proto tato kapitola pojednává o převodu obrazu na černobílý, tzn. o prahování obrazu a cannyho detekci hran.

2.2.1 Prahování

Pokud chceme převést obraz víceúrovňového jasu (v odstínech šedi) na obraz o dvou jasech (černá a bílá), využijeme metodu prahování, nazývanou také jako Thresholding. Určí se prahová hodnota a bodům pod touto úrovní je přiřazena hodnota 0, bodům nad touto hodnotou hodnota 1. Případně se místo hodnoty 1 přiřazuje hodnota 255. Prahování se dělí do dvou skupin a to na globální a lokální. U globálního prahování je hodnota prahu určena pro celý obraz. U lokálního prahování se tato hodnota prahu mění, v různých částech obrazu je hodnota prahu různá (Dobeš, 2008). Ukázka prahování obrazu je na obrázku 3.

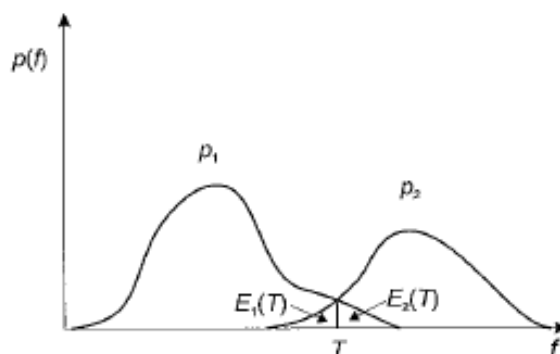


Obrázek 3: Prahování - vlevo původní snímek, vpravo výsledek prahování (Dobeš, 2008).

Hodnotu prahu lze získat několika způsoby. Jeden z nich je dle výběru hodnot jasu z obrazu. Další způsob je pomocí histogramu, kde zkoušením různých hodnot prahu nalezneme takový, který dává nejlepší výsledky. Tato metoda je ale úspěšná jen v interaktivním prostředí, kde uživatel může měnit hodnoty prahu a hned vidí výsledek histogramu. Dále existují metody, kde hodnota prahu se určuje automaticky, bez pomoci uživatele. (Dobeš, 2008)

Metoda automatického nalezení prahu je možná v případě rozumného rozdělení jasu v obraze. Jedná se o obrazy, kde obraz má na histogramu dva výrazné vrcholy. Jasové hodnoty jednoho vrcholu odpovídají pozadí. Hodnoty v oblasti druhého vrcholu odpovídají popředí, což jsou body, které nás zajímají.

Další metodou je způsob nalezení optimálního prahu, nazývána také jako Otsuova metoda (obrázek 4). Zde se jedná o to, že histogram je považován za funkci hustoty pravděpodobnosti. Předpokládá se, že histogram lze rozdělit na dvě různé křivky, které nám určí popředí a pozadí, viz. obrázek 4. Hlavním cílem této metody je nalezení takového prahu, aby byly tyto křivky co nejdál od sebe. (Dobeš, 2008)



Obrázek 4: Prahování - hledání optimálního prahu (Dobeš, 2008).

Na ukázkou obrázek 5, kde je zobrazeno využití lokálního prahování.



Obrázek 5: Prahování - výsledek lokálního prahování (Dobeš, 2008).

2.2.2 Cannyho hranová detekce

Jedna z metod pro detekci hran v obraze je Cannyho funkce. Myšlenka této metody je založená na hledání filtrem skokovou hranu. U 2D obrazu si skokovou hranu můžeme představit jako schod. Při pohledu na tři kritéria je detektor vhodný pro skokové hrany: (Hlaváč, Sedláček, 2000)

1. detekční kritérium - významné hrany nesmí být přehlédnuty, jedna hrana nesmí mít vícenásobnou odezvu
2. lokalizační kritérium - rozdíl mezi skutečnou a nalezenou polohou musí být minimální
3. požadavek jedné odezvy - detektor na jednu hranu nesmí reagovat vícenásobně, tento požadavek je částečně zajištěn prvním kritériem, je nutný hlavně pro zašuměné a nehladké hrany

Algoritmus Cannyho hranového detektoru:(Hlaváč, Hledání hran, 2016)

- najdi přibližně směry gradientu
- pro každý pixel nalezni 1D derivaci ve směru gradientu pomocí optimální masky spojující vyhlazení a derivaci
- najdi lokální maxima těchto derivací
- hranové body získkej prahováním s hysterezí
- proved syntézu hran získaných pro různě velká vyhlazení

Cannyho hranový detektor se používá ve většině aplikací, kde je potřeba nalézt hrany obrazu. Implementace tohoto detektoru je dostupná.(Hlaváč, Hledání hran, 2016)



Obrázek 6: Cannyho detektor hran. (MathWorks, 2014)

2.3 Metody rozpoznávání obrazu

Metod na rozpoznávání obrazu je velké množství. Kdyby se v této kapitole měly popisovat všechny kapitoly, tato práce by nebyla o ničem jiném. Proto v této kapitole je jen pár vybraných metod, ze kterých se dále bude vybírat, které by se mohly využít. Jedná se o metody FLANN - Knihovna přibližného rychlého vyhledávání pro porovnání nejbližších sousedů, FAST - Features from accelerated segment test, SIFT - Scale-Invariant feature transform, SURF - Speed-Up Robust Features.

2.3.1 FLANN - Knihovna přibližného rychlého vyhledávání pro porovnání nejbližších sousedů

FLANN je knihovna pro rychlé vyhledávání nejbližšího souseda ve vysokých dimenzionálních prostorech. Tato knihovna obsahuje mnoho algoritmů. Systém vybere vždy ten nejlepší algoritmus podle optimálních parametrů v závislosti na datovém souboru. (FLANN - Fast Library for Approximate Nearest Neighbors, 2012)

Hledání nejbližšího souseda je jedním ze zásadních významů různých aplikací. Například pro rozpoznávání obrazu, kompresi dat, vzor rozpoznávání a klasifikaci, strojové učení, statistiky a analýzy dat. Řešení tohoto problému na vysoce dimenzionálních prostorech je velmi obtížné a neexistuje žádný algoritmus, který by našel výrazně lepší řešení než standardní řešení hrubou silou. To vedlo k většímu zájmu o algoritmy pro hledání nejbližšího souseda. Tyto algoritmy se ukázaly jako zlepšení pro nejpraktičtější aplikace a ve většině případů je aplikací těchto algoritmů řešení rychlejší a vyhledávání přesnější. (Muja, 2009)

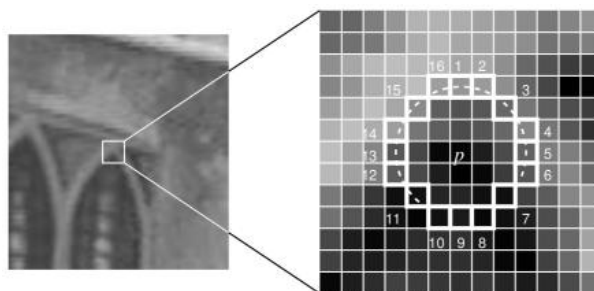
Knihovna FLANN je napsána v programovacím jazyce C++. FLANN se může jednoduše použít i v programovacím jazyce C, Pythonu a v prostředí MATLAB.

2.3.2 FAST - Features from accelerated segment test

Je to metoda rozpoznávání, kterou v roce 2006 navrhl Edward Rosten a Tom Drummond. Tato metoda je rychlejší než jiné metody jako například SIFT a SUSAN.

Postup rozpoznávání obrazu (FAST Algorithm for Corner Detection, 2014):

1. Vyber bod p a rozhodni, zda je klíčový bod obrazu. Intenzita tohoto bodu je I_p .
2. Vyber vhodnou prahovou hodnotu t .
3. Vyber kruh s 16 body pro porovnání.



Obrázek 7: Metoda FAST - výběr kruhu (FAST Algorithm for Corner Detection, 2014).

4. Nyní existuje hodnota prahového bodu p a kolem něj množina n sousedících bodů v kruhu. Tyto sousedící body jsou všechny buď jasnější než je součet intenzity bodu p a jeho prahové hodnoty t nebo jsou všechny tmavší než rozdíl intenzity pixelu p a jeho prahové hodnoty t .
5. Pomocí vysokorychlostního testu je doporučeno vyloučení velkého množství neprahových bodů. Tento test se zabývá pouze čtyřmi body, a to 1, 9, 5, 13. První dva, 1 a 9 jsou testovány, zda jsou světlejší nebo tmavší. Pokud ano, pak jsou testovány pixely 5 a 13. Když bod p je prahový, pak nejméně 3 z nich musí být světlejší nebo tmavší. Pokud ani jeden z nich tomuto požadavku neodpovídá, pak p nemůže být prahový bod.

První tři body této metody jsou řešeny přístupem strojového učení. Další dva body jsou řešeny pomocí potlačení lokálních maxim velikosti gradientu.

Strojové učení detektoru prahu

Postup (FAST Algorithm for Corner Detection, 2014):

1. Vyber množinu z obrázků pro trénování.
2. Spust FAST algoritmus na všechny obrázky pro nalezení klíčových bodů.
3. Pro každý klíčový bod ulož do vektoru hodnoty 16 bodů kolem něj. Udělej to pro všechny obrázky a dostaneš příznakový vektor P .
4. Každý bod z těchto 16 bodů může nabývat tří stavů. Může být tmavší, světlejší nebo podobný.

$$S_{p \rightarrow x} = \begin{cases} d, & I_{p \rightarrow x} \leq I_p - t \\ s, & I_p - t < I_{p \rightarrow x} < I_p + t \\ b, & I_p \leq I_{p \rightarrow x} \end{cases}$$

d - pro tmavější s - pro podobné b - pro světlejší

5. V závislosti na těchto stavech rozděl vektor P do 3 podmnožin (P_d, P_s, P_b)
6. Definuj logickou proměnnou K_p , pokud je p prahový a jinou, pokud není prahový.
7. Použij ID3 algoritmus (rozhodovací strom) a dotazuj se na všechny podmnožiny pomocí K_p pro poznatky ohledně skutečné třídy. To vybere x , které poskytne více informací o tom, zda je bod p rohový. Měřeno entropií K_p .
8. Tento krok se rekurzivně provede ve všech podmnožinách, pokud je entropie nulová.
9. Takto vytvořený rozhodovací strom se používá pro rychlou detekci i v jiných obrazech.

Potlačení lokálních maxim velikosti gradientu

Detekce více klíčových bodů v přilehlých lokalitách je další problém. Ten je řešen pomocí potlačení lokálních maxim velikosti gradientu.

Postup (FAST Algorithm for Corner Detection, 2014):

1. Vypočítej výsledek funkce v pro všechny detekované feature body. V je suma absolutního rozdílu mezi hodnotou bodu p a 16 okolními body.
2. Vezmi dva sousední klíčové body, vypočítej jejich v hodnoty.
3. Smaž ten s nižší hodnotou v .

2.3.3 SIFT - Scale-invariant feature transform

Je to metoda publikována Davidem G. Lowe poprvé v roce 1999, ve zdokonalené verzi poté publikována roku 2004. Metoda je založená na rozpoznávání množství jednoznačně určitelných částí (klíčových bodů), které vedou k úspěšnému rozpoznání objektu. Tyto jednoznačně určitelné části se označují jako Image Features. Každá z Image Features je popsána vektorem o velikosti 128 hodnot, které jasně identifikují daný Image Features v rámci rozpoznávaného obrazu. Jelikož je tato metoda náchylná na šum a změny jasu či různě velké obrazy nebo jinak pootočené, je potřebné, aby funkce, které hledají potřebné Image Features a vytváří z nich deskriptory v obraze, byly vůči tomuto odolné. Také musí zaručit, aby nalezené Image Features odlišily dané objekty od sebe. (Kolomazník, 2014)

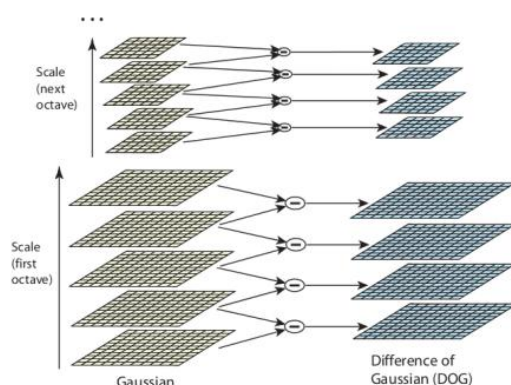
Dříve než začne proces hledání Image Features, je potřeba daný obraz poupravit, abychom zamezili velké náročnosti na paměť počítače. Musí se použít filtr pro zobrazení obrazu ve stupních šedi. Postup hledání klíčových deskriptorů je v jednoduchosti znázorněn zde:

1. Detekování extrémů v scale-space - Načtení obrazu ve stupních šedi a následné nalezení všech hledaných bodů.
2. Zpřesnění polohy klíčového bodu - Pro každý bod se určí měřítko, jeho umístění a ověří se jeho stabilita.
3. Přiřazení orientace klíčovým bodům - Pro každou lokaci bodů se určí její směr. Každý bod může nabývat jednoho nebo více směrů.
4. Sestavení deskriptorů klíčových bodů - Pro všechny nalezené bodů se dopočítávají popisné vektory jejich okolí.

Detekování extrémů v scale-space

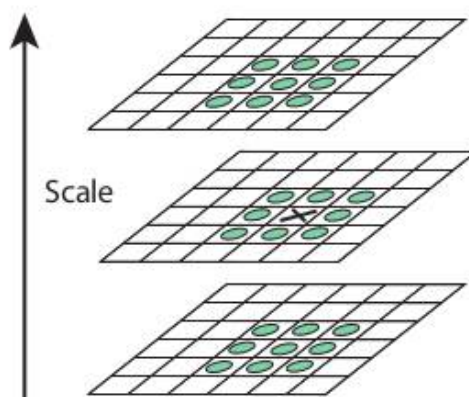
V prvním kroku zpracování obrazu se používá filtr Scale-space, abychom detekovali vhodné features. Je potřeba nalézt takové features, které lze detekovat opakovaně a nezávisle na měřítku, úhlu pohledu a intenzitě světla v obraze. Vhodná místa (body) jsou nalezená maxima v derivačním obraze $D(x,y,\sigma)$, který se získává metodou DOG (Difference-of-Gaussian). Tato metoda je pro aproximace derivace v obraze pomocí rozdílu dvou Gaussově rozostřených obrazů.

Použije se tedy Gaussovo rozmazávání na vytvořenou posloupnost obrazů (oktáva). Z posloupnosti k obrazů se vytvoří postupně k-1 rozdílových obrazů DOG. Poté, co se dokončí zpracování jedné oktávy, přejde se na další. Tento cyklus se opakuje do té doby než se dosáhne rozumné velikosti obrazu. Zrychlit výpočet algoritmu lze tím, že první obraz nové oktávy lze odvodit z předchozí oktávy a jejich posledních dvou obrazů. Celé toto zpracování obrazu pomocí Gaussova rozmazávání je vidět na obrázku 8. (Kolomazník, 2014)



Obrázek 8: Metoda SIFT - znázornění oktáv (Introduction to SIFT, 2015).

Dále je potřeba nalézt lokální maxima a minima v $D(x,y,\sigma)$. Každý takový bod se srovnává s jeho osmi sousedy v aktuálním snímku. Srovnává se také s devíti sousedy, kteří se nacházejí ve stejné oktávě, ale před a po aktuálním snímku. Srovnávaný bod je vybrán jen tehdy, pokud je větší nebo menší než všech jeho 26 sousedů. (Kolomazník, 2014)



Obrázek 9: Metoda SIFT - hledání lokálního maxima (Introduction to SIFT, 2015).

Zpřesnění polohy klíčového bodu

Jakmile jsou lokalizovány potencionální klíčové body, musí se z nich získat přesnější výsledky. Zkoumá se jejich okolí, zda rozdíl mezi jeho hodnotou a okolními hodnotami sousedů je velký. Odstraní se tímto krokem body, které jsou málo kontrastní či body náchylné na šum. Také je vhodné odstranit body nacházející se na hranách, jelikož tyto body jsou špatně lokalizovatelné.

Původně byla pozice features považována za pixel. Později se začaly pozice features vypočítávat na subpixelovou přesnost pomocí proložení bodu a jeho okolí 3D kvadratickou funkcí. Poloha této features je určena pomocí minima této funkce. (Kolomazník, 2014)

Přiřazení orientace klíčovým bodům

V dalším kroku je potřeba přidat každému vzorku klíčového bodu histogram obsahující směr a velikost. Jelikož tyto hodnoty jsou využity ve výpočtu výsledných deskriptorů, je zaručeno, že tyto deskriptory budou invariantní vůči natočení i vůči velikosti. (Kolomazník, 2014)

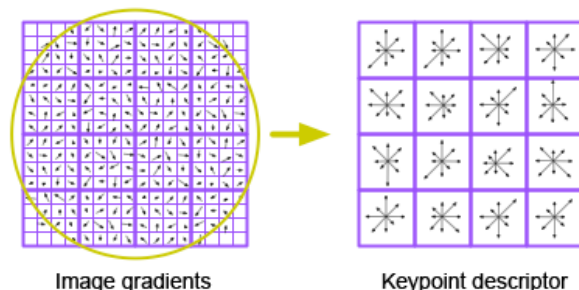
Výpočet probíhá při rozostřeném obrazu a první derivace se spočítá v osách x a y (d_x, d_y). Výpočet velikosti pro každou features:

$$m(x, y) = \sqrt{d_x^2 + d_y^2}$$

Výpočet orientace pro každou features:

$$\Omega(x, y) = \arctan d_y^2 d_x^2$$

Sestavení deskriptorů klíčových bodů

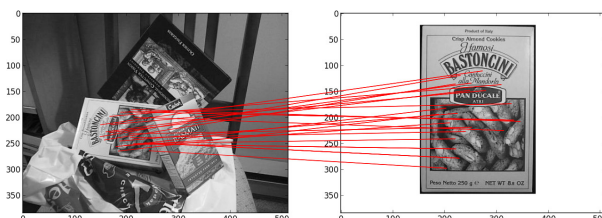


Obrázek 10: Metoda SIFT - výpočet keypoint descriptoru

Je vytvořen deskriptor klíčových bodů. Okolo klíčového bodu je přiřazeno 16 x 16 sousedů. Ty jsou rozděleny do 16 dílčích bloků o velikosti 4x4, viz obrázek 10. Pro každý podblok je vytvořeno 8 orientovaných histogramů. Celkem je k dispozici tedy 128 hodnot. Ty jsou prezentovány jako vektor pro vytvoření klíčových deskriptorů. Kromě toho bylo provedeno i opatření, aby byly omezeny účinky osvětlení a rotace na výsledný deskriptor. Toho se dosáhne normalizací vektoru na jednotkovou délku, kterou se právě změna jasu a kontrastu v okolí potlačí. (Introduction to SIFT, 2015)

Porovnávání klíčových bodů

Posledním krokem je porovnávání klíčových bodů mezi objekty. Klíčové body mezi dvěma objekty jsou porovnávány pomocí jejich deskriptorů uložených v databázi. Pro hledání nejlepších kandidátů na shodu mezi objekty se používá metrika euklidovské vzdálenosti. V některých případech se může stát, že shoda mezi dvěma deskriptory je moc blízko, může nastat vlivem rušení či jiných důvodů. V tom případě se vezme druhý nejbližší deskriptor. Jestliže tento druhý deskriptor není výrazně vzdálený od prvního deskriptoru, je nalezená shoda neplatná. Tímto mechanismem eliminace je omezeno množství nalezených shod (až 90 %), které jsou neplatné. V případě, kdy se tímto způsobem identifikují alespoň tři features, můžeme mluvit o silném důkazu přítomnosti objektu ve zkoumaném obraze. (Introduction to SIFT, 2015)



Obrázek 11: Metoda SIFT (General introduction to SIFT_{PyOCL}., 2014).

2.3.4 SURF - Speeded-Up Robust Features

Metoda SURF byla představena v roce 2006 a jejím hlavním záměrem bylo zkoustruovat výpočetně rychlý a stabilní deskriptor pro práci v reálném čase. Inspirací byla metoda SIFT, která vykazovala velice dobré výsledky, ale byla pomalá. Zrychlení metody SURF oproti metodě SIFT bylo dosaženo v oblasti sestavení měřítkově nezávislé reprezentace obrazu (scale-space) a v redukci velikosti výsledných deskriptorů. (Hrúz, 2015)

Detekce klíčových bodů

SIFT scale-space je aproximován rozdíly Gaussových funkcí. Jelikož tyto funkce mají velké odezvy i v okolí hran obrazu, je nutné tyto body odebrat z množiny klíčových bodů pomocí determinantu Hessianovy matice. Rychlý výpočet determinantu je prováděn pomocí integrálního obrazu. (Hrúz, 2015)

Integrální obraz

Je to struktura vybudovaná nad vstupním obrazem umožňující rychlý výpočet součtu hodnot, které jsou uvnitř libovolné obdélníkové oblasti vstupního obrazu. Tento součet může být vyjádřen:

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (1)$$

$I(i, j)$ - vstupní obraz

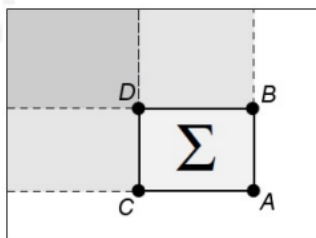
$I_{\Sigma}(x, y)$ - integrální obraz

Pak lze součet hodnot obdélníkového regionu uvnitř vstupního obrazu dosazením hodnot vypočítat pomocí vzorce:

$$\Sigma = A - B - C + D \quad (2)$$

Σ - hledaný součet

A, B, C, D - hodnoty integrálního obrazu v daných souřadnicích podle obrázku 12.



Obrázek 12: Metoda SURF - Výpočet součtu hodnot v obdélníkovém regionu Σ pomocí integrálního obrazu (Hrúz, 2015).

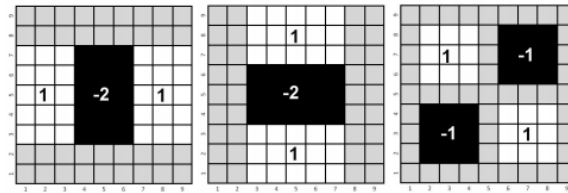
Hessianova matice

Hessianova matice bodu (x,y) s měřítkem σ je definována:

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{yx}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix} \quad (3)$$

$L_{xx}(x, y, \sigma)$ - hodnota druhé parciální derivace podle x konvoluce vstupního obrazu $I(x,y)$ s Gaussovou funkcí $G(x, y, \sigma)$ a měřítkem σ v bodě (x,y)

Hessianova matice v metodě SURF používá pro výpočet prvků obdélníkové funkce, které aproximují druhou derivaci v diskrétní formě. Jedná se o konvoluci obrazu s filtry, což je znázorněno na obrázku 13. Implementace integrálního obrazu umožňuje velice rychlé vypočítání odezvy na dané obdélníkové filtry, což lze vidět opět z obrázku 13. (Hrůz, 2015)



Obrázek 13: Metoda SURF - Aproximace druhé derivace gaussovy funkce podle osy x , osy y a xy (Hrůz, 2015).

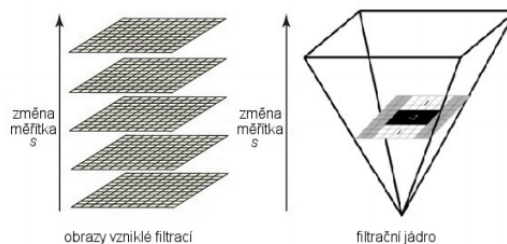
Filtry na obrázku mají velikost $9 \times$ pixelů. Tyto filtry odpovídají nejmenšímu možnému uvažovanému měřítku během sestavování scale-space metody SURF. Obrazy, které vzniknou filtrací jsou označovány D_{xx} , D_{yy} a D_{xy} a determinant Hesianu se vypočte pomocí vzorce:

$$Det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (4)$$

Abychom mohli mít pro všechny měřítka konstantní váhu w , odezvy na obdélníkové filtry jsou normalizovány vzhledem ke své velikosti. Pro nejpřesnější aproximaci Gaussova jádra je použita váha w a je stanovena na hodnotu 0,9. (Hrůz, 2015)

Scale-space

Při hledání klíčových bodů je důležité, aby byly nalezeny i při změně měřítka obrazu. Je tedy nutné, aby se takovéto body hledaly uvnitř měřítkově nezávislé reprezentace. Proto se v této metodě scale-space konstruuje postupně z jednotlivých obrazů vzniklých filtrací. Jelikož se implementuje integrální obraz, je složitost konstantní a nedochází k nárůstu výpočetní složitosti. Proto ve scale-space větších měřítek dosahuje konvolucí vstupního obrazu s filtračním jádrem zvyšujících se rozměrů. Jelikož se v této metodě kaskádovitě negenerují scale-space, je tato metoda rychlejší než metoda SIFT. Obraz také není potřeba podvzorkovat a nedochází k aliasingu. (Hrůz, 2015)



Obrázek 14: Metoda SURF - Konstrukce scale-space (Hrúz, 2015).

Scale-space se dělí na několik oktáv. Základ první oktávy je obraz, který vznikne aplikací filtračních jader, které mají rozměr 9×9 . V případě zvětšujícího se měřítka filtračního jádra se jejich rozměr zvětšuje na hodnoty 15×15 , 21×21 a 27×27 . U tohoto měřítka se splňuje podmínka, aby nejvyšší vrstva byla dvojnásobně větší než vrstva první. L_0 se označuje krok, o který jednotlivé jádra narůstají. Tento krok byl stanoven na $\frac{2}{3}$ velikosti základního jádra. Potom platí, že $l_0 = 6$. (Hrúz, 2015)

Aby byla vytvořena následující oktáva, je pro to důležitá druhá vrstva předchozí oktávy. Zároveň při sestavování této oktávy je důležitý i krok l_0 z předešlé oktávy, který se zdvojnásobí. Druhá oktáva je vytvořena odezvami filtračního jádra o velikosti 15×15 , 27×27 , 39×39 a 51×51 . Hodnota l_0 je pak $l_0 = 2 \times 6 = 12$.



Obrázek 15: Metoda SURF - změna měřítka (Hrúz, 2015).

Z odezev na filtraci je postupně vypočítána hodnota determinantu Hessiánské matice, která je důležitá pro detekci klíčových bodů. Z těchto hodnot determinantu získáme obraz $H_{Det}(x, y, \sigma)$. Tyto obrazy představují výslednou měřítkově nezávislou reprezentaci vstupního obrazu. V této reprezentaci jsou pak lokalizovány klíčové body. (Hrúz, 2015)

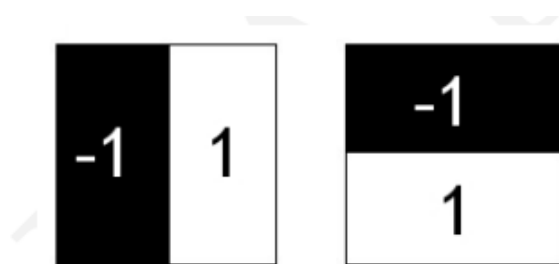
Detekce maxim uvnitř scale-space

Klíčové body jsou detekovány jako maxima ve scale-space, které se hledají stejným způsobem jako v metodě SIFT. Body se tedy porovnávají se svými 26 sousedy v rámci scale-space (8 sousedů + 18 sousedů z vyšší a nižší vrstvy). Jestliže má zkoumaný bod ve svém okolí nejvyšší hodnotu, je považován za klíčový bod. (Hrúz, 2015)

Přiřazení orientace klíčovým bodům

Stejně jako u metody SIFT i metoda SURF přiděluje každému klíčovému bodu jeho orientaci. Výsledný deskriptor je pak invariantní vůči rotaci. V metodě SIFT se využívá histogram orientací, v metodě SURF se zkoumá odezva Haarové vlnky v kruhovém okolí daného bodu. Kruhové okolí má střed v klíčovém bodě a poloměr je roven $6s$, kde s značí měřítko vrstvy scale-space, ve kterém se klíčový bod nachází. Aby bylo umožněno zapojení integrálního obrazu do procesu výpočtu odezvy, je Haarova vlnka aproximována obdélníkovými filtry ve směru osy x i y . Tyto filtry jsou měřítkově závislé na velikosti $4s \times 4s$. (Hrúz, 2015)

D_x a d_y jsou označovány výsledky konvoluce obrazu s Haarovou vlnkou. Tyto odezvy jsou váženy koeficienty Gaussova kruhového okna. Vážené hodnoty d_x a d_y se zakreslí do souřadnicového systému a výsledná orientace klíového bodu se určí jejich součtem v rámci výseče, která pokrývá $\pi/3$ prostoru. Ve výsečích poté vzniknou vektory definované $\sum d_x$ a $\sum d_y$ a výseč. V této výseči se nachází největší z vektorů, který představuje orientaci klíového bodu. (Hrúz, 2015)

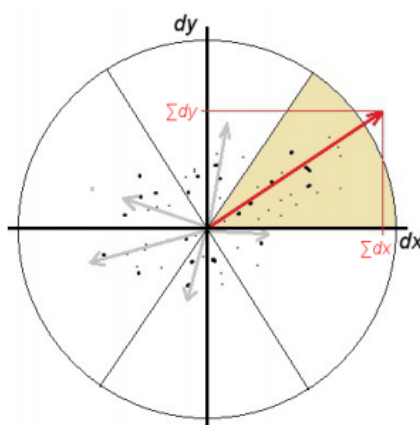


Obrázek 16: Metoda SURF - Haarova vlnka. (Hrúz, 2015)

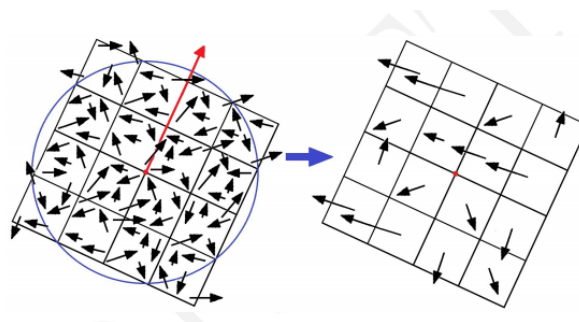
Sestavení deskriptorů klíčovými body

Pro sestavení deskriptorů klíčovými body se prvně vytvoří čtvercové oblasti kolem klíového bodu. Tyto oblasti se středem v klíčovém bodě jsou natočeny podle orientace daného klíového bodu a její délka hrany je $20s$. Čtvercové oblasti se dále rozdělují na 4×4 podoblasti. Pro každou oblast se určuje pět pravidelně rozmístěných bodů, pro které se vypočítává odezva na Haarovu vlnku v obou osách (x, y). Stejně jako v předchozím kroku se i zde odezvy pětice bodů získávají aplikací obdélníkových filtrů. Jediný rozdíl je v tom, že rozměr filtračních jader je $2s \times 2s$ a nemůže se využívat integrální obraz. Opět jsou odezvy označeny jako d_x a d_y a také jsou váženy koeficienty Gaussova kruhového okna pro zlepšení výsledného deskriptoru. (Hrúz, 2015)

V každé podoblasti se spočítají $\sum d_x$ a $\sum d_y$ a pro odolnost deskriptoru na změny osvětlení v obraze jsou vypočítány i absolutní hodnoty $\sum |d_x|$ a $\sum |d_y|$. Pro každou podoblast v okolí klíového bodu se vytvoří vektor $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Výsledný deskriptor se skládá z takovýchto vektorů pro všech 16 podoblastí. Okolí klíového bodu se popisuje 64 biny (16 podoblastí \times 4-prvkový vektor v). (Hrúz, 2015)



Obrázek 17: Metoda SURF - Určení orientace, největší vektor leží ve výseči (žlutá barva), která určuje výslednou orientaci klíčového bodu (Hrúz, 2015).



Obrázek 18: Metoda SURF - Vlevo je odezva na Haarovu vlnku v okolí klíčového bodu dle orientace a napravo výsledný deskriptor (Hrúz, 2015).

2.4 Umělé neuronové sítě

Umělé neuronové sítě jsou matematické modely. Tyto neuronové sítě jsou založeny na biologické neuronové síti. Základní prvek neuronové sítě je neuron, který má základní vlastnost učit se. Znalost je ukládána pomocí vah jednotlivých vazeb. Neuronové sítě jsou využitelné v případě, že nejsme schopni popsat matematické vztahy v daném procesu. Můžeme je také využít, pokud matematický model sestavíme a zároveň je řešení tohoto matematického modelu složité.

Dynamika neuronové sítě se dělí do tří skupin. První skupina je organizační, kde je topologie sítě. Další je adaptivní, tam se neuronová síť učí a poslední skupina je aktivní, kde se síť vybavuje. (Maceček, 2013)

V praxi se neuronové sítě používají všude tam, kde nevadí malá nepřesnost, jelikož úspěšnost neuronových sítí je 95 %. Nesmí vadit ani náročnost algoritmu ze strany finanční (hardware) a časové (nutnost rychlého rozhodnutí). Využití neuronových sítí můžeme rozdělit do několika oblastí: (Olej, 2010)

1. problémy aproximace funkcí,
2. klasifikace do tříd, klasifikace situací,
3. řešení predikčních problémů (vývoj cen akcií, změny kurzu, optimalizace rezervace jízdenek, predikce počasí),
4. problémy řízení procesů,
5. transformace a analýza signálu,
6. rozpoznávání a případná rekonstrukce obrazců (zkoumání otisků, pospisů),
7. asociáční problémy, simulace paměti

2.4.1 Neuron

Neuron je základní jednotka neuronové sítě. Neuron má n vstupů x_1, x_2, \dots, x_n . Každý vstup je ohodnocen synaptickou vahou w , která reprezentuje citlivost, s jakou vstup působí na výstup. Váhy mohou být kladné či záporné. Každý neuron má svou prahovou hodnotu Θ , která určuje, kdy je neuron aktivní a kdy pasivní. Pokud je hodnota vstupu neuronu nižší než prahová hodnota, na výstupu signál neuronu je odpovídající pasivnímu stavu. Pokud se prahová hodnota neuronu překročí, stane se neuron aktivní a výstup neuronu roste až do určité maximální hodnoty. Tato maximální hodnota je dána oborem hodnot aktivační funkce f . (Olej, 2010)

V neuronové síti nejsou jenom prahy, které byly znázorněny výše, ale každý neuron má jeden zvláštní práh s vahou w_0 , kterému je přiřazena hodnota 1. Pak příspěvek všech vstupů do neuronu je označen jako u a pro lineární bázové funkce platí vzorec:

$$u = w_0 + \sum_{i=1}^n w_i x_i$$

Nelineární aktivační funkce f pak má vstup, který se skládá z příspěvků všech vstupů "u" a výstup této funkce je výstup neuronu:

$$y = f_a(u)$$

Dále se mohou používat jako aktivační funkce jiné nelineární funkce, které mají takovou vlastnost, že v okolí prahové hodnoty se blíží jednotkovému skoku. (Ondráček, 2006)

2.4.2 Struktura neuronové sítě

Struktura neuronové sítě určuje, jaký je počet neuronů a jejich vzájemné propojení. Jsou rozlišovány neurony vstupní, skryté a výstupní. Stav neuronové sítě určují stavy neuronů a jejich synaptické váhy všech spojení určují konfiguraci neuronové sítě. Rozlišují se dva typy neuronových sítí, a to dopředná a rekurentní. Dopředná neuronová síť je taková, kde se signál šíří jen jedním směrem, dopředu. (Olej, 2010)

U rekurentních neuronových sítí se signál šíří v kruhu, jelikož skupina neuronů je zapojená do kruhu. Synapse těchto sítí jsou orientovány různými směry a neuron může být současně jak vstupní, tak výstupní. Rekurentních neuronových sítí využívá tehdy, pokud použití dopředné neuronové sítě nebylo efektivní či nebylo úspěšné. Tyto sítě jsou zároveň dynamicky složitější než dopředné neuronové sítě. Učení těchto sítí je možné jak s učitelem, tak bez učitele. Nejjednodušší zástupce rekurentních neuronových sítí učení s učitelem jsou Hopfieldovy sítě, o nichž se píše v kapitole 2.5. (Olej, 2010)

2.4.3 Učení neuronové sítě

Učení neuronových sítí se dělí do dvou skupin. Učení s učitelem a učení bez učitele.

V případě učení s učitelem se rozhodovací pravidlo učí na základě trénovací množiny, která poskytuje správnou informaci o požadovaném výstupu neuronové sítě. Tato trénovací množina má dvojice prvků, kde jsou vstupy a jejich očekávané výstupy. Množina všech těchto hodnot představuje správnou část chování systému. Množina všech dostupných dat není jen v trénovací množině, ale data jsou rozdělena na trénovací a testovací množinu. Poměr prvků v testovací množině a trénovací množině je přibližně 70 - 80 % dat v trénovací množině a zbytek v testovací množině. Někdy je možné mít data rozdělena na tři části, tou třetí je validační množina. Průběžně se periodicky vyhodnocuje chyba (energie) jak na trénovací množině, tak na tréninkové množině a pokud tato hodnota validace nějakou dobu neklesá, je adaptace zastavena. (Učení s učitelem, 2016)

Výpočet chybové funkce:

$$E_t = \sum_{k=1}^m [\hat{y}_k(t) - y_k]^2, (Olej, 2010)$$

t - čas,

m - počet vzorů trénovací množiny

k - pořadové číslo vzoru trénovací množiny

y_k - požadované hodnoty na výstupu z neuronové sítě

$\hat{y}_k(t)$ - skutečné hodnoty výstupu z neuronové sítě

Učení probíhá v několika krocích (Kačenka, 1998):

- - Inicializace vah na malé náhodné hodnoty
- - předložení množiny vstupů a jejich odpovídajících výstupů
- - výpočet aktuálního výstupu
- - úprava vah v případě, že výstup se liší od očekávaného výstupu
- - ukončení nebo opakování od kroku 2 v případě, že výsledek není ten, který je očekáván

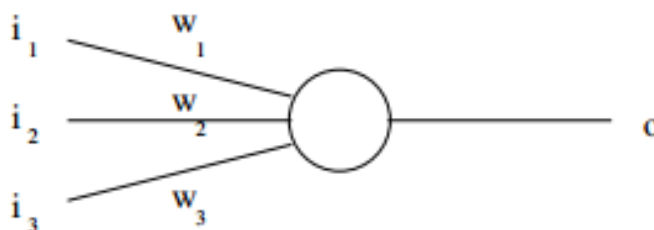
Učení s učitelem se využívá například u lineární neuronové sítě, dopředné neuronové sítě, radiální bázové funkce, pravděpodobnostní neuronové sítě, prediktivní adaptivní rezonanční teorie a jiné. (Olej, 2010)

Učení bez učitele se použije všude tam, kde nemáme k dispozici žádnou trénovací množinu, a proto nemůžeme předat informace učiteli. Používá se pro analýzu dat. Pozorovaná data se vysvětlují pomocí matematického modelu buď se statistickým přístupem, což je učení statistického modelu z dat. Druhý přístup je deterministický, kde se učí podle jiných měř podobnosti dat, například podle vzdálenosti. U učení bez učitele se tedy používá místo trénovací množiny výstup z klasifikátoru. Učení bez učitele je důležité, pokud není předem známa klasifikace dat a pokud by klasifikace dat člověkem byla finančně náročná. (Hlaváč, Učení bez učitele, 2016)

Učení bez učitele je u neuronových sítí založeno na schopnosti rozeznat ve vstupech stejné nebo podobné vlastnosti a třídit je do skupin podle daných vlastností. Podobné vlastnosti se sdružují do map či shluků. Algoritmus učení nemá žádnou informaci o tom, jak by měl vypadat požadovaný výstup neuronů. Neuronové síti se poskytují jen vstupní hodnoty. Princip tohoto algoritmu je výpočet vzdálenosti mezi vzory a aktuálními hodnotami. Hledají se minimální vzdálenosti mezi těmito vzory. (Olej, 2010)

Učení bez učitele se využívá například u asociativní paměti, Kohonenovy samoorganizující se mapy, adaptivní rezonanční teorie a jiných. (Olej, 2010)

2.5 Hopfieldova síť



Obrázek 19: Umělé neurony použité v Hopfieldové síti.

Hopfieldova síť má strukturu sítě takovou, že každý neuron je vzájemně spojený s každým jiným neuronem. Je to jednovrstvá, rekurzivní a binární síť. Hopfieldovou síť je možno použít jako asociativní paměť či k řešení optimalizačních problémů. Hopfieldova síť s asociativní paměti se využívá pro obrazový vstup. Využitelnost Hopfieldovy sítě není pro spojité vstupy, jelikož jsou jednotlivým pixelům obrazu přiřazovány binární hodnoty. (Šnorek, 2002)

2.5.1 Uspořádání Hopfieldovy sítě

Hopfieldova síť se skládá z tolika neuronů, kolik má vstupů. U obrázků to znamená, že každému pixelu odpovídá jeden neuron. Všechny neurony jsou do sítě připojeny pouze jedním vstupem. Výstup každého neuronu vede do všech dalších neuronů pomocí synaptické váhy w_{ij} , a tímto se vytváří zpětnovazební smyčky. Takto vzniká symetrická síť, která má diagonálně symetrickou matici vah. Váhy jsou totiž v obou směrech stejné ($w_{ij} = w_{ji}$). Na vstupy neuronů se nepřivádí jejich vlastní výstup ($i \neq j$). Hodnoty vah na diagonále jsou nulové.

U Hopfieldovy sítě je práh θ nulový a aktivační funkce se neposouvá. Všechny signály (vstupní, stavové, výstupní) mají binární hodnoty. Používají se buď hodnoty 1 nebo 0 nebo bipolární hodnoty +1 a -1. Jakmile do neuronu přijdou vstupní signály, provede se jejich vážený součet a jejich výsledek se použije pro aktivační funkci. Výstupem z daného neuronu je poté hodnota aktivační funkce. Jelikož výstupy se přivádějí na vstupy ostatních neuronů, je potřeba, aby se dodržely pravidla o hodnotách jak aktivační funkce, tak signálů, a proto zase musí nabývat binárních hodnot. (Šnorek, 2002)

Jelikož Hopfieldova síť patří mezi asociativní paměti, jejím výstupem je přímo nalezený vzor, který odpovídá předložnému vzoru. V případě, že bychom chtěli Hopfieldovu síť použít jako klasifikátor, museli bychom přidat ještě jednu vrstvu, která by řešila klasifikaci. (Šnorek, 2002)

2.5.2 Učení a vybavování Hopfieldovy sítě

Učení Hopfieldovy sítě probíhá algoritmem, který je uveden pod tímto odstavcem. Pro každý vstupní obraz (vzor) se vytvoří matice o rozměru $N \times N$, kde N je počet vstupů (pixelů). Vynásobením i -tého a j -tého vstupu (tzn. metoda násobení každý s každým) se vytvářejí jednotlivé prvky matice. V případě, že jsou vstupy stejné ($i = j$), hodnota tohoto pixelu je nulová. Tímto procesem se vytvoří symetrická matice s hodnotami obsahující 0 a 1 v případě binární reprezentace či s hodnotami -1 a 1 v případě bipolární reprezentace a na hlavní diagonále jsou pouze nulové hodnoty. Potřebná výsledná matice vah vznikne součtem všech vstupních matic daných vzorů. (Šnorek, 2002)

Postup učení Hopfieldovy sítě (Šnorek, 2002):

- Nastavení vah podle vstupních vzorů

$$w_{ij} = \begin{cases} \sum_{s=0}^{M-1} x_i^s x_j^s, & \text{pro } i \neq j, \\ 0, & \text{pro } i = j, 0 \leq i, j \leq N - 1 \end{cases}$$

w_{ij} - váha mezi neuronem i a j x_i a x_j - i -tý (j -tý) prvek s -tého vstupního vzoru s hodnotou 0 či 1 (bipolární +1 a -1)

- Opakování učícího se procesu - V případě, že nebyly předloženy všechny trénovací vzory, vracíme se na krok 1, v opačném případě učení Hopfieldovy sítě ukončíme

Po naučení všech vstupních vzorů lze přejít na vybavování Hopfieldovy sítě. Hopfieldova síť se od ostatních umělých neuronových sítí liší tím, že učení probíhá jednorázovým učením jednoho vzoru, ale u vybavování už je tomu jinak.

Při předložení vzoru pro vybavování se postupně stavy mění, jak je zmíněno v algoritmu vybavování sítě níže. V každém kroku se tyto stavy pravděpodobně změní. Tento algoritmus se opakuje tak dlouho, pokud se hodnota žádného neuronu nezmění ve dvou krocích za sebou. To znamená, že nynější a předcházející stav je shodný. Tento stav se prohlásí za výstup sítě a je shodný s některým ze vstupních vzorů. Princip porovnávání vzorů u Hopfieldovy sítě je založen na Hammingově metrice. Proto jako správnou odpověď Hopfieldova síť bere ten vzor, který má tuto metriku nejmenší. (Šnorek, 2002)

Vybavování v Hopfieldově síti (Šnorek, 2002):

- Inicializace stavů - nastavení počátečních stavů dle předloženého vzoru,

$$\mu_i(0) = x_i, 0 \leq i \leq N - 1$$

$\mu_i(t)$ - stav i -tého neuronu v čase t x_i - prvek vzoru

- Iterace až do nalezení odpovědi -

$$\mu_i(t + 1) = f \left[\sum_{j=0}^{N-1} w_{ij} \mu_j(t) \right], 0 \leq i \leq N - 1$$

Tento krok se provádí do doby, dokud se stavy neuronů nemění. Jinak řečeno, rozdíl mezi těmito a předchozími stavy je nulový. Konečné výstupy neuronu jsou přímo jednotlivými body výstupního obrazce.

- Opakování procesu - po nalezení odpovědi můžeme vložit nový vzor a celý postup opakovat

2.5.3 Hebbovo adaptační pravidlo

Myšlenka Hebbova adaptačního pravidla je taková, aby váhové hodnoty, které jsou na spojení mezi dvěma neurony a jsou současně ve stavu on či off narůstaly. Pokud tato spojení mezi neurony nejsou současně ve stavu on či off, budou tyto váhové hodnoty klesat. Jak lze vidět, změna synaptické váhy je závislá na souhlasné aktivitě spojených dvou neuronů, tedy součinu jejich stavů. Změna váhové hodnoty na spojení mezi dvěma neurony (u_i, u_j) je dána vztahem:

$$\Delta w_{ij} = \alpha * y_i * y_j,$$

α - koeficient učení

y_i, y_j - stavy neuronů u_i a u_j

Neuronové sítě s asociativní pamětí mají váhové hodnoty determinovány tak, aby si mohly zapamatovat množinu asociovaných vzorů. Všechny asociace jsou tvořeny párem složeným ze vstupního a výstupního vektoru. Pro asociované neuronové sítě lze váhové hodnoty na daných spojích nalézt pomocí Hebbova adaptačního pravidla. Lze tedy říci, že změna synaptické váhy spojení mezi dvěma neurony je závislá na souhlasné aktivitě, tedy součinu stavů mezi dvěma neurony. Opačná aktivita tuto vazbu mezi dvěma neurony oslabuje. Může nastat i to, že jsou vstupní vzory ortogonální a dochází k jejich vzájemnému rušení a tudíž tato síť se není schopná tato spojení naučit. (Volná, 2014)

2.5.4 Vlastnosti Hopfieldovy sítě

V případě použití Hopfieldovy sítě jako asociativní paměti jsou dvě hlavní omezení. Prvním omezením je počet vzorů pro učení, který je poměrně malý. V případě naučení sítě na větší množství vzorů, může síť konvergovat k nějakému úplně jinému obrazci, který není mezi vzory. Pro zamezení tomuto stavu je nutné si spočítat, aby počet vzorů byl menší než 15 % bodů vstupního obrazu. Například, pokud budeme mít 10 vzorů, musí mít každý vzor alespoň 67 bodů. Pro lepší názornost a jistotu raději 70 bodů. Poté každý vstupní obraz může mít například velikost 7×10 pixelů. Váhová matice bude o velikosti $70 \text{ times } 70$ a bude obsahovat 4900 vah. Druhé omezení pro Hopfieldovou síť je, že lze pracovat pouze se vstupními vzory, které si nejsou moc podobné. (Šnorek, 2002)

Jednou z hlavních nevýhod Hopfieldovy sítě je náročnost na paměť. Tento nárok narůstá kvadraticky s počtem vstupů. Další nevýhodou je, že tato síť není odolná vůči natočení či posunutí vstupního obrazu. Pokud není vstup naprosto stejně umístěn (posun i natočení), Hopfieldova síť jej nejspíše nerozpozná.

Jednou z hlavních výhod Hopfieldovy sítě je automatické naučení inverzních vzorů k naučeným vzorům. V případě, že bychom dali na vstup Hopfieldovy sítě inverzní obraz, po několika opakováních dostaneme odpovídající vzor. (Šnorek, 2002)

2.6 Vícevrstvá neuronová síť

U vícevrstvé neuronové sítě se musí zdůraznit, že je dopředná, tudíž signál jde pouze jedním směrem a vylučují se cykly. Neurony jsou v těchto sítích uspořádány do vrstev. (Malý, 2007)

Výstup každého neuronu v dané vrstvě je přiváděn na vstupy všech neuronů následující vrstvy. Jsou to takzvané spoje mezi neurony, tedy orientované čáry spojující neurony v dané vrstvě s neurony v následující vrstvě. Synaptická váha w_{ij} asociuje spoj neuronu i v dané vrstvě s neuronem j v následující vrstvě.

Potenciál neuronu j se počítá:

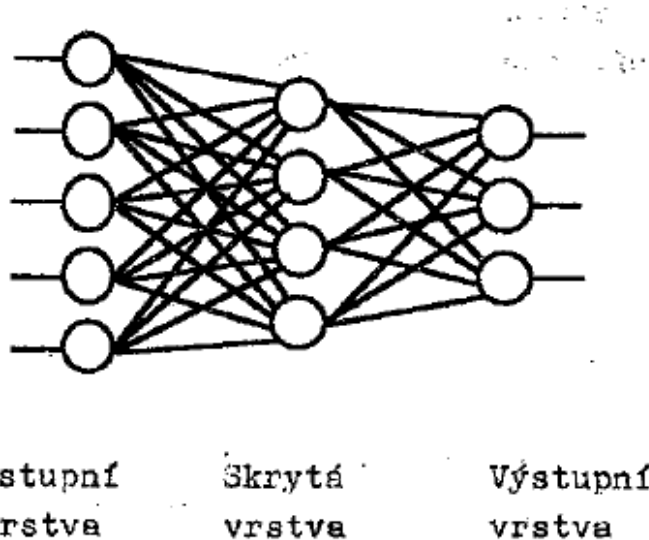
$$\xi_j = \sum_{i=0}^n w_{ji}y_i, (Mal, 2007)$$

n - počet neuronů v předcházející vrstvě

y_0 - jedničkový vstup zvážený w_{j0}

y_i - výstupní hodnoty neuronů v předcházející vrstvě

Neurony jsou rozděleny v rámci sítě na vstupní, skryté a výstupní. Vstupní neurony slouží pouze pro vstupní signál do sítě. Hodnoty neuronů se inicializují hodnotami složek vstupního signálu a nepodílí se na výpočtu. Skryté neurony slouží pro přenos a transformaci původního signálu, který se naposledy zpracuje výstupními neurony. Výstupní hodnoty výstupních neuronů reprezentují výsledek výpočtu neuronové sítě. (Malý, 2007) Ukázka vícevrstvé neuronové sítě je na obrázku 20.



Obrázek 20: Vícevrstvá neuronová síť (Bíla, 1995).

Speciální případ vícevrstvé sítě je síť jednovrstvá, kde jsou jen vstupní a výstupní neurony. Ve skutečnosti se jedná o dvě vrstvy, ale mluví se o ní jako o jednovrstvé, jelikož vstupní vrstva se v tomto kontextu vypouští, protože se nejedná o plnohodnotné neurony. Tyto neurony nevypočítávají žádný výstup, ale jsou to pouze vstupní data. Obecně řečeno, pokud mluvíme o m -vrstvé síti, ve skutečnosti mluvíme o síti, která má $m + 1$ vrstev. Schématické označení této sítě udává celkový počet vrstev a počet neuronů v nich a vstupní vrstva se nevynechává. Třívrstvá síť je znázorněna schématickým označením 3-4-3-2. (Malý, 2007)

2.6.1 Učení vícevrstvé neuronové sítě

Nejčastěji se využívá učení s trénovací množinou, kde má množina dané vstupní vzory a jejich očekávané výstupy. Učící algoritmy vícevrstevných neuronových sítí se snaží minimalizovat globální chyby odezvy neuronové sítě na jednotlivé vzory z trénovací množiny. Mezi algoritmy patřící k učení vícevrstvé neuronové sítě pro čení jednotlivých neuronů patří například od Rosenblatta *Perceptron learning algorithm* nebo Gallantův *Pocket algorithm*. Pro učení celé sítě se používají algoritmy Back Propagation, Levenberg-Marquardt či Quick Propagation. (Ondráček, 2006)

2.6.2 Backpropagation

Backpropagation je adaptační algoritmus zpětného řízení chyby. Zároveň je nejpožívanějším adaptačním pravidlem u vícevrstevných neuronových sítí s dopředným šířením signálu (ve směru od vstupní vrstvy k výstupní vrstvě). Tento algoritmus se skládá ze tří etap. První etapa je dopředné šíření vstupního signálu tréninkového vzoru, druhá etapa je zpětné šíření chyby a poslední, třetí etapa je aktualizace váhových hodnot na spojích mezi neurony. Tímto postupem se získá odezva neuronové sítě na vstupní podnět vybuzením neuronů vstupní vrstvy. (Volná, 2014)

Pro neuronové sítě s adaptační metodou učení backpropagation musí mít aktivační funkce následující vlastnosti: funkce musí být spojitá, diferencovatelná, monotónně neklesající. Proto se velice často používá sigmoida a hyperbolický tangens jako aktivační funkce.

Celý proces s adaptační metodou backpropagation se opakuje do té doby dokud není splněna podmínka ukončení, tzn. chyba neuronové sítě E je přijatelná. (Volná, 2014)

3 Metodika

K tvorbě softwaru pro tuto diplomovou práci budou použity dva programy. Visual Studio 2013, ve kterém budu programovat část pro rozpoznávání mincí obrazem. Modul v tomto programu bude naprogramován v jazyce C++ a budou využívat i knihovnu OpenCV. Druhým programem je Control Web 7. Ten použiji pro ovládání různých prvků modelu třídění a také pro vytvoření uživatelské aplikace. Pro učení neuronové sítě bude ještě využit program Matlab.

3.1 Visual Studio

Visual Studio je produkt firmy Microsoft. Je to bohaté vývojové prostředí pro různé aplikace, například na mobilní zařízení s operačním systémem Windows, Android a iOS, desktopové aplikace pro operační systém Windows, moderní webové aplikace a cloudové služby. Jsou zde nástroje pro různé velké projekty. Podporuje agilní vývoj software, což znamená, že si můžeme v backlogu nastavovat různé úkoly, plánovat sprinty či vytvářet různé grafy a řídicí panely. Další z funkcí, kterou tento software podporuje je verzování v Git, který nejvíce využívají větší firmy. Ale lze jej také využít pro vlastní vývoj, kde chceme mít znát historii aplikace. V neposlední řadě podporuje i testovací nástroje, jako například průběžné testování v DevOps, ale i jednodušší manuální a průzkumové testování. (Visual Studio, 2016)

Visual Studio je zpracován ve více verzích. Pro software k diplomové práci budu využívat verzi Profesional 2013, která má školní licenci.

3.1.1 OpenCV

Ve Visual Studiu je podporována knihovna OpenCV. Tato knihovna je šířena s BSD licencí, což znamená svobodný software. Lze jí využívat zdarma i pro komerční účely. Knihovna má C++, C, Python a Java rozhraní a podporuje Windows, Linux, iOS a Android. OpenCV je navrženo pro výpočetní efektivitu se silným důrazem na rychlost běhu aplikace. Do knihovny přispívá více než 47 tisíc uživatelů. Rozsah využití knihovny je obrovský, například pro zpracování obrazu, moderní robotiku a jiné aplikace. (OpenCV, 2016)

3.2 Control web

Control web je programový systém s širším využitím. Firma Moravské přístroje vytvořila tento systém aby nebyl omezen jen na vizualizaci a sběr dat na operátorských pracovištích, ale umožnil vytvářet různé aplikace pro přímé řízení strojů v reálném čase a uměl vytvářet komplexní aplikace v oblasti simulace a modelování a zpracování dat v počítačových sítích. Tento systém byl navržen pro řízení jednotek strojů, umí propojovat výrobní technologii s informačním systémem nebo může být využit jako datový server pro webových klientů. Lze jej také využít pro modelování a simulaci různých procesů. (Programový systém Control Web, 2016)

Programového systému Control web má několik verzí. Vývojová verze je určena pro tvorbu a testování aplikací, obsahuje okno integrovaného vývojového prostředí (IDE), ve kterém je aplikace vytvářena. Veškeré změny, které se dějí v grafickém režimu, se automaticky generují i do textové podoby aplikace a naopak. Mimo okna IDE se při vývoji využívají také okna Palety nástrojů a Inspektora přístrojů. Paleta přístrojů obsahuje všechny komponenty, které lze do aplikace přetáhnout myší. Parametry přístrojů lze upravovat pomocí Inspektora přístrojů, což je speciální okno, které zobrazuje všechny vlastnosti komponenty. Ve vývojové verzi lze upravovat jen soubory *.cw*. Před uvedením do ostrého režimu je aplikace přeložena do podoby *.cwx*. Tento soubor je určen pro druhou verzi systému Control Web, a to pro Runtime. Runtime verze se používá zejména pro rychlejší spouštění aplikace, má rovněž menší nároky na paměť. (Programový systém Control Web, 2016)

3.3 Matlab

Matlab je využíván mnoha inženýry a vědci pro analyzování, návrh a následnou simulaci systému. Využívá se v automobilových systémech, v meziplanetární kosmické lodi, v sledovacích zdravotních zařízeních, v inteligentních rozvodných sítích či LTE sítích mobilních operátorů. Další využití je pro strojové učení, zpracování signálů, zpracování obrazů, počítačové vidění, komunikaci, řízení designu, robotiku apod. Matlab má mnoho grafických možností pro zobrazování získaných dat, a tím i pro lepší poznatky z těchto dat pro uživatele. (MathWorks, 2014)

V mé práci bude Matlab využíván pro učení vícevrstvé neuronové sítě. Rozhodla jsem se pro jeho využití, protože jsou jeho funkce optimalizované, a tím zaručují rychlejší chod aplikace. Výsledky z vícevrstvé neuronové sítě budu pak zpracovávat v jazyce C++.

3.4 Snímací kamera

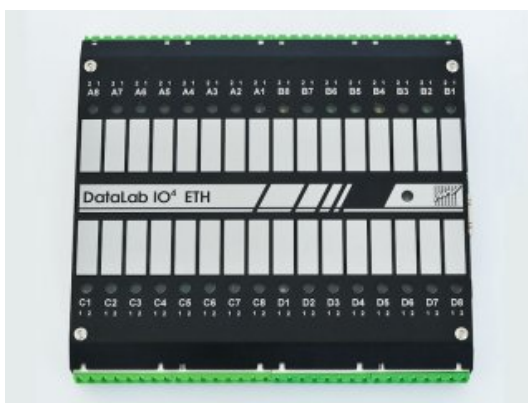
Pro snímání mincí bude využita kamera Microsoft LifeCam Studio s FullHD. Tato webová kamera má snímač s vysokým rozlišením 1080p, a tím zaručuje kvalitní videa. Precizní skleněný širokoúhlý objektiv umožní natáčet videa s velkou přesností. Kamera má technologii TrueColor, která přináší jasná a barevná videa za všech světelných podmínek. A dále i technologii ClearFrame, která zajišťuje plynulý a detailní obraz. Kamera rovněž obsahuje kvalitní mikrofon pro přenos audia, ale to v mé práci využito nebude. (Hardware: LifeCam Studio, 2016)



Obrázek 21: Webová kamera Microsoft LifeCam Studio (Hardware: LifeCam Studio, 2016)

3.5 Datalab IO

Datalab IO je průmyslově vstupně/výstupní jednotka nové generace určená pro spolupráci s počítačem. Pro počítač zprostředkovává měření i nastavování veličin. Datalab je k dispozici s komunikačním rozhraním USB, síťovým rozhraním Ethernet nebo sériovým rozhraním RS-485. Koncepce jednotek je modulární. Každá jednotka má 1, 2 nebo 4 pozice pro speciální moduly k využití dle potřeb aplikace. Tyto pozice lze kombinovaně zaplnit moduly například pro 8 logických vstupů s moduly pro 8 logických výstupů s relé nebo výstupními tranzistory s otevřeným kolektorem. Mohou se také použít čítačové moduly, moduly pro teploměrná čidla anebo kombinované moduly, které obsahují například digitální vstupy a výstupy na jediném modulu. V případě, že je potřeba využít více modulů než jednotka dovoluje, je umožněno využít najednou více jednotek k počítači. (Modulární systém průmyslových vstupů/výstupů DataLab IO, 2009)



Obrázek 22: Jednotka Datalab IO (Modulární systém průmyslových vstupů/výstupů DataLab IO, 2009)

4 Praktická část

V praktické části této práce je popsán průběh jejího vytváření. Několik kapitol se vztahuje k hardwarové části, kde je například řešen krokový motorek a jeho driver. V další části jsou popsány úpravy aplikace vyvíjené v Control webu. Velká část této kapitoly spadá pod rozpoznávání obrazu, kde je popsána implementace vybraných metod a také jejich výsledky.

4.1 Seznam rozpoznávaných měn

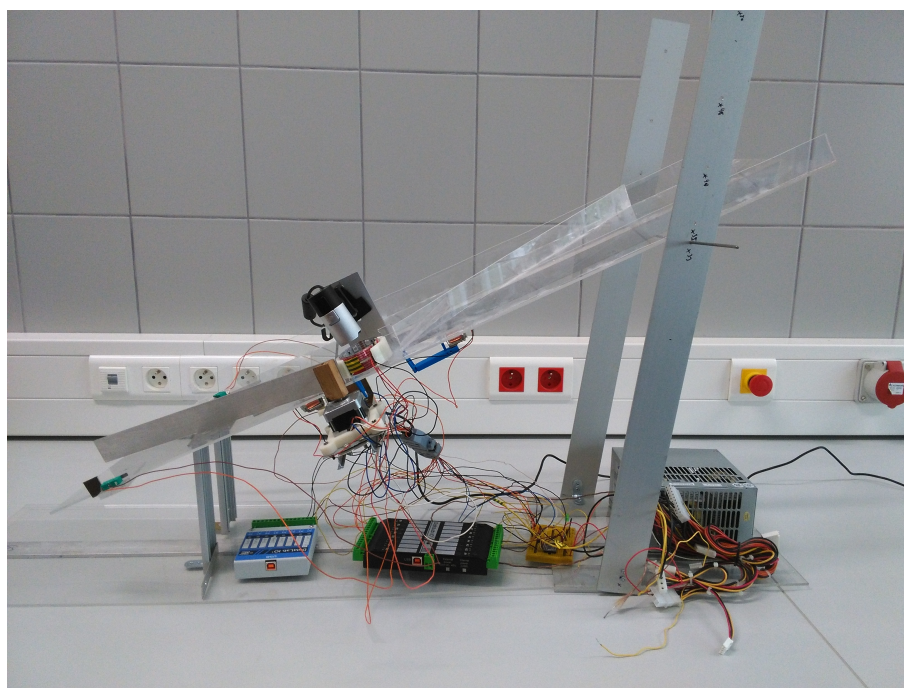
V této práci jsem se rozhodla využít všechny měny, které nabízí ke směně Česká národní banka. Jedná se o 32 různých měn z celého světa. Seznam těchto měn je uveden v tabulce níže, seznam všech mincí v těchto měnách i s jejich rozměry je popsán v příloze 6.

Měna	Zkratka	Měna	Zkratka
Americký dolar	USD	Jihoafriický rand	ZAR
Australský dolar	AUD	Jihokorejský won	KRW
Brazilský real	BRL	Kanadský dolar	CAD
Britská libra	GBP	Maďarský forint	HUF
Bulharský lev	BGN	Malajský ringgit	MYR
Česká koruna	CZK	Mexické peso	MXN
Čínský juan	CNY	Norská koruna	NOK
Dánský koruna	DKK	Novozélandský dolar	NZD
Euro	EUR	Polský zlotý	PLN
Filipínské peso	PHP	Rumunské nové leu	RON
Honkongský dolar	HKD	Ruský rubl	RUB
Chorvatská kuna	HRK	Singapurský dolar	SGD
Indická rupie	INR	Švédská koruna	SEK
Indonéská rupie	IDR	Švýcarský frank	CHF
Izraelský šekel	ILS	Thajský baht	THB
Japonský jen	JPY	Turecká lira	TRY

Tabulka 2: Seznam mincí

4.2 Model pro třídění mincí

Model pro třídění mincí bylo potřeba vyrobit znovu, jelikož některé jeho části bylo nutné pro snímání obrazu zlepšit. To znamenalo vyřešit osvětlení tak, aby na nasvícené minci byl reliéf zřetelný. K tomu jsem použila speciální osvětlovací kruh, který nasvítí minci ze všech stran. Více o tomto osvětlení je napsáno v samostatné kapitole 4.8. Kvůli konstrukci osvětlení se muselo snížit ohraničení modelu v zuženém místě pro snímání mincí. V tomto prostoru je ohraničení sníženo tak, aby pod osvětlením, které je na něm položeno, prošla i ta nejtlustší mince, která má být vytríděna. Další změnou oproti předchozí verzi bylo použití jiného krokového motoru. V bakalářské práci jsem použila pro model krokový motor ze staré disketové mechaniky. Krokový motor použitý v novém modelu má jednotlivé kroky přesnější než krokový motorek z dřívějšího modelu. Více o použitém krokovém motoru čtete v kapitole 4.5. K tomuto krokovému motoru se musel vyrobit i speciální držák, kterým bude připevněn ke konstrukci modelu. K modelu jsem přidala i spínací kontakty pro automatické zjišťování, zda se rameno krokového motoru nachází na konci požadovaného rozsahu otáčení krokového motoru a má se zastavit. Dále ještě do modelu bylo přidán konektor, který zajišťuje jeho jednodušší přenositelnost. Uživatel má možnost rozpojit obvod mezi modelem a nepájivým polem s Datalabem a model přenést. Zbylé části modelu zůstaly nezměněny.



Obrázek 23: Třídící model.

4.3 Napájení modelu

Napájení modelu je řešené počítačovým zdrojem. Využívá se jeden čtyřpinový periferní výstup, který má 2x uzemnění (černý vodič), 1x vodič s napětím +5 V (červený) a 1x vodič s napětím +12 V (žlutý). Žlutý vodič je využíván pro napájení osvětlení, pro relé v propouštěcích bránách, pro spínací kontakty a také výstupní piny driveru ke krokovému motoru. Červený vodič je využíván pouze pro napájení vstupní části driveru krokového motoru. Ostatní piny, které jsou vyvedeny z počítačového zdroje, nejsou využívány.

4.4 Elektromagnety a brány pro průchod mincí

Zužený prostor pro zajištění průchodu jen jedné mince v prostoru pro snímání kamerou nebylo potřeba a zůstal stejný jako u modelu z bakalářské práce. Tento prostor zcela vyhovoval. Ke změně došlo u typu relé. V bakalářské práci se použily dvě různé relé a jelikož jedno z nich bylo napájeno méně než 12 V napětím, bylo potřeba přidat stabilizátor napětí a blokační kondenzátory pro správnou funkci. V novém modelu už jsou použité dvě relé s napětím 12 V. Z tohoto důvodu musely být upraveny rozměry bran, které na tyto relé navazují. Před vstupem do tohoto zúženého prostoru nebyl dříve řešen průchod dvou mincí položených na sobě. Zde už je to vyřešeno malým plastovým dílem, který vstup do zúženého prostoru postupně omezuje výškou. Tím je zajištěno, že se do prostoru dostane jen jedna mince. Zapojení těchto bran do obvodu je znázorněno v příloze na obrázku 49.

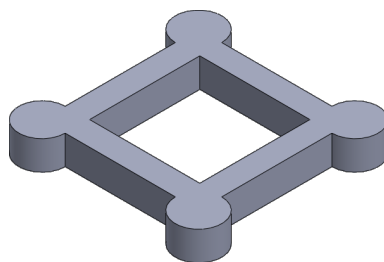
4.5 Krokový motor SX16-0402N

Pro ovládání třídícího ramene byl tuto práci vybrán krokový motor značky Microcon s označením SX16-0402N. Krokový motorek, který byl využit v bakalářské práci nebyl vhodný, jelikož u něj nebyla možnost jemného nastavení chodu motoru a muselo se zmenšování kroku řešit ozubenými kolečky. Ale ani tak se nedosáhlo tak malého kroku, jakého lze docílit s krokovým motorem SX16-0402N. Jeden krok tohoto motoru je $1,8^\circ$. Tento krokový motor se skládá ze 4 vodičů. Na jednom vinutí je modrý a bílý vodič, na druhém vinutí oranžový a černý vodič. V případě prohození těchto vodičů by motor nefungoval. Dle katalogu jsem zjistila tyto technické parametry ke krokovému motoru:

Technický parametr	Hodnota
Statický moment	0.11 Nm
Jmenovitý proud	0.4 A
Indukčnost	32 mH
Odpor	30Ω
Zbytkový moment	0.012 Nm
Moment setrvačnosti rotoru	20gcm ²
Hmotnost	0.19 kg

Tabulka 3: Technické parametry krokového motoru SX16-0402N

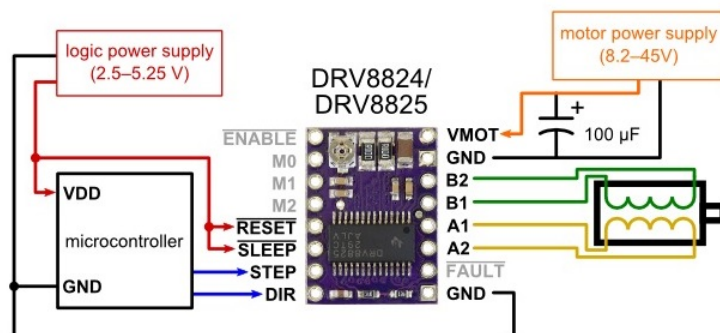
Pro připojení krokového motoru k modelu bylo potřeba sestavit speciální držák. Ten se skládá ze dvou malých konzolí, hliníkové lišty a držáku motoru vytištěném v 3D tiskárně. Tento držák vytištěný v 3D tiskárně je na obrázku 24. Držák musel být konstruován tak, aby v něm motor pevně držel. Každý menší pohyb způsobí nepřesnost v používání motoru. Jak vypadá držák motoru je vidět v příloze..



Obrázek 24: Držák krokového motoru.

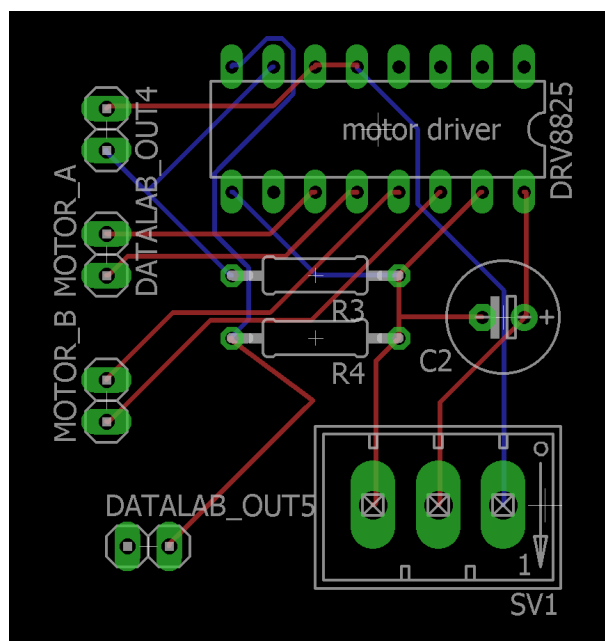
4.6 Řízení krokového motoru

Řízení krokového motoru Microcon SX16-0402N není vůbec složité. Při správném zapojení je potřeba ovládat jen dva vstupy. Na driveru sloužícím pro ovládání krokového motoru se nastavují jen dva vstupy. A to vstup pro určení kroku (STEP) a druhý pro určení směru otáčení motoru (DIR). V případě, že chceme, aby se motor pohnul o jeden krok, dáme na vstup STEP jedničku. Pro více kroků za sebou na vstup STEP opakovaně vkládáme 0 a 1, kde 0 znamená stop a 1 další krok. Na vstupu DIR ovládáme směr otáčení motoru, a to vložením jedničky na tento vstup pro otáčení po směru hodinových ručiček. Vložením nuly na tento vstup dáme určit, že chceme motorem otáčet opačným směrem. V případě potřeby zastavení motoru a následném udržení polohy musíme na vstupu STEP dát nulu. Zapojení obvodu krokového motoru s jeho driverem je vidět na obrázku 25. Ovládání pinů STEP a DIR probíhá jednotkou Datalab. Driver je více popsán v teoretické části v kapitole 2.1.3.



Obrázek 25: Zapojení krokového motoru pomocí driveru DRV8825. (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

Toto zapojení je aktuálně provedeno v nepájivém poli. Řešení však není bezpečný, jelikož uživatel může omylem obvod přepojit, a tím by došlo ke zkratování některého prvku. Proto jsem navrhla desku plošných spojů, která by po vytištění a napájení tento problém vyřešila.



Obrázek 26: Deska plošných spojů zapojení krokového motoru. (Using the DRV8825 Stepper Motor Driver Carrier, 2016)

4.7 Spínače pro detekci zastavení motoru

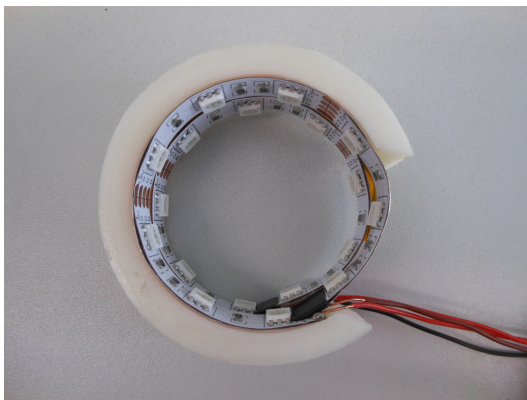
Krokový motor, který je zde využit, má jako každý jiný krokový motor otáčení 360° kolem své osy. V této práci je ovšem nezbytné, aby se motor otáčel jen ve vyhrazeném rozsahu. V případě otočení mimo tento rozsah by mohlo dojít k poškození modelu. Pro detekci, zda je krokový motor na jednom z konců požadovaného rozsahu jsem zvolila mikropsínače P-B172B. Oba jsou napájeny na napětí 12 V. V případě, že rameno připojené na krokový motor dojde na konec požadovaného rozsahu, přitlačí na mikropsínač a otáčení motoru se vypne. Zapojení spínačů je znázorněno na obrázku 49 v příloze.

4.8 Osvětlení

Pro správné rozpoznávání mincí je potřeba mít pořádné osvětlení. Tímto se předejde problémům s denním světlem či případně světlem ze stropu, které by při snímání kamerou dělalo stíny. Třídící model měl v dřívější verzi osvětlení pomocí bílé LED pásky dlouhé 5 cm. To bylo umístěno přibližně 7 cm pod plexisklem pod prostorem pro snímání mince. Toto osvětlení sloužilo pouze pro jednodušší měření rozměru mince. V novém modelu je nahrazeno jiným osvětlením, které minci nasvítí správně jak pro měření rozměru, tak pro rozpoznávání obrazu mince.

Nové osvětlení je složeno z LED pásky, ale tentokrát to jsou 2 červené LED pásky o délce 15 cm. Tyto pásky jsou zatočeny do kruhu a jsou vůči sobě trochu pootočené, aby nebyly LED diody na pásce nad sebou, ale vedle sebe. LED pásky jsou vsazeny do kruhového držáku, který byl namodelován v programu SolidWorks a poté vytisknut na 3D tiskárně. Toto osvětlení je umístěno kolem snímání mince a napomáhá k lepšímu osvětlení reliéfu mince. Inspirací pro toto osvětlení byl osvětlovač DataLight firmy Moravské přístroje. Pro mou práci však není vhodný, jelikož je příliš velký a nevejde se do modelu. Jeho šířka je 225 mm a délka 273 mm. Oproti tomu můj osvětlovač má průměr jen 60 mm. I váhově jsou rozdílné, jelikož DataLight váží 350 g, moje osvětlení jen 13 gramů. To je velká výhoda pro můj model, jelikož nemusí být tak zatížen a nemusí se řešit jeho zpevnění. U DataLight je možnost výběru barvy osvětlení, lze volit bílé, červené, modré nebo zelené. To však bych u mého modelu nevyužila. Já jsem zvolila pro osvětlení výraznou červenou barvu. Na obrázku 27 je zobrazena konstrukce osvětlení. V příloze na obrázku 48 je vidět finální umístění osvětlení do modelu. Zapojení LED pásek v elektrickém obvodu je znázorněno na obrázku 49 v příloze.

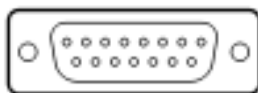
Osvětlení na modelu je napájeno na napětí 12 V. Osvětlovací prvek je zapnutý pouze při třídění mincí z důvodu úspory energie. Teorii k LED osvětlení naleznete v mé bakalářské práci (Čupová, 2014).



Obrázek 27: Držák LED osvětlení.

4.9 Konektor

Model pro třídění mincí obsahuje nemalé množství elektronických prvků. Pro jednoduché přenášení modelu, aniž by se zároveň musela přenášet jednotka Datalab či nepájivé pole, jsem zvolila možnost rozpojení obvodu pomocí konektoru. Zamezí se případnému rozpojení obvodu v nepájivém poli či utržení nějakého elektronického prvku z modelu při přenášení. Pro tento model byly zvoleny konektory Canon s 15 piny. Aby byl tento konektor bezpečnější a nevytrhávaly se z něj vodiče, je uložen ve speciálním krytu. Níže je zobrazeno rozložení vodičů na pinech konektoru.



Obrázek 28: Konektor Canon s 15 piny.

PIN	využití	PIN	využití
1	Červená LED +12 V	9	Krokový motor bílý vodič
2	Červená LED +12 V	10	Krokový motor modrý vodič
3	Červená LED zem	11	Krokový motor červený vodič
4	Bílá LED +12 V	12	Krokový motor černý vodič
5	Bílá LED zem	13	Spínací tlačítko +12 V
6	Relé +12 V	14	Spínací tlačítko zem
7	Relé +12 V	15	Spínací tlačítko zem
8	Relé zem		

Tabulka 4: Rozdělení pinů konektoru

4.10 Uspořádání pořadí mincí

Při své bakalářské práci jsem řadila mince na výstupu z modelu dle toho, jaká je pravděpodobnost jejich výskytu, aby nejbližší počátku byly mince, které jsou nejvíce využívány. Tehdy jsem měla jen 3 měny na třídění, takže jsem mohla vyrobit zatřídovací krabičky na míru pro každou měnu. Jelikož nyní chci třídít 32 měn, vytvořit pro každou měnu sadu krabiček by bylo nesmyslné a pro měny s velkým množstvím mincí ani možné. Proto je jejich uspořádávyjí dle hodnoty od nejmenší po největší. U měn, které mají 7 více mincí, jsem se musela rozhodnout pro třídění dvou hodnot mincí do jedné krabičky. A to z toho důvodu, že model má omezenou šířku a zatřídovací krabičky by byly moc malé a mince by do nich nevešly. Řešením by bylo mít konec modelu rozšířený o alespoň 12 cm. Dalším řešením by bylo třídění těchto měn dvoukrokově a to prvně roztřídit jak bylo popsáno vyše, dva druhy mincí do jedné krabičky. Pak obsah krabiček, které mají dvě hodnoty mincí vysypat zpět na vrchol modelu a znovu třídít jen tyto mincí. To by však podstatně prodloužila celková doba třídění.

4.11 Rozpoznání mincí v aplikaci

Vybraných 32 měn obsahuje velké množství mincí a jejich rozpoznávání je proto hodně složité. Hlavní problémem jsou stejné rozměry různých mincí, kterých je poměrně dost. Pokud jsou tyto rozměry stejné u různých měn, pak to není problém, pokud chceme mince třídít v rámci jedné měny. Může nastat problém, když se mince třídí podle rozměrů v rámci jedné měny a někdo do modelu vloží i minci jiné měny, která má stejný rozměr jako nějaká mince tříděné měny. V tom případě se mince zatřídí špatně. Proto je nutné mince rozpoznávat nejen pomocí rozměru, ale také obrazem. Při třídění mincí jednotlivých měn je už větší problém s množstvím mincí stejných rozměrů. Tyto už se musí kontrolovat obrazem, aby se mince správně zatřídily. Proto třídění může trvat poměrně déle než třídění jen v rámci jedné měny.

Nejvíce stejných rozměrů bylo u velikosti mince 20,00 mm (10 centavos(BRL), 5 stotinek(BGN), 1 Kč(CZK), 25 centavos(PHP), 10 líp(HRK), 50 rupií(IDR), 1 jen(JPY), 1 rand(ZAR), 1 baht(THB)), u velikosti 20,50 mm (2 dolary(AUD), 20 stotinek(BGN), 5 Jiao(CNY), 50 líp(HRK), 10 centů(NZD), 50 groszů(PLN), 10 bani(RO), 1 rubl(RUB), 10 korun(SEK), 25 kurus(TRY)), u velikosti 23,00 mm (50 centavos(BRL), 5 Kč(CZK), 5 rupií(INR), 100 rupií(IDR), 10 šekelů(ILS), 2 randy(ZAR), 2 pesos(MXN), 1 dolar(NZD), 1 zlotý(PLN), 2 ruble(RUB), 50 centů(SGD)).

Při rozpoznávání mincí pomocí obrazu je problémem, že měření rozměru pomocí kamery není přesné. Mince jsem měřila v pixelech s tolerancí. Pro každý možný rozměr jsou v souboru uloženy druhy mincí těchto rozměrů, které budou rozpoznávány.

Rozpoznávání mincí pomocí obrazu má ještě jednu velkou výhodu. U měření rozměru se nedá rozpoznat, zda je v modelu vložena mince a například knoflík. Při rozpoznávání obrazem se hned zjistí o jakou minci se jedná, případně pokud jej aplikace nezná, vyřadí ji do speciální krabičky.

4.12 Aplikace v Control web

V aplikaci vytvořené v programu Control web bylo provedeno pár změn oproti verzi v bakalářské práci. Například výběr měny pro třídění se prováděl pomocí Radio buttonu, ale nyní pro větší množství měn (32) jsem zvolila Select box. Vedle Select boxu je ještě zobrazena vlajka státu vybrané měny. Také výběr pro třídění mincí mezi měnami musel projít změnou. Uživatel si v Select boxu kliknutím vybere měny, které chce třídít. Seznam měn je zobrazen v tabulce pod Select boxem. Uživatel může zvolit maximálně 5 měn. Tyto druhy měn se uloží v popiscích, u kterých se zobrazují i počet nalezených mincí této měny. V dřívější verzi aplikace se rozměr mince měřil přímo v Control webu. Nyní pro rychlejší třídění se toto zpracovává v aplikaci pro rozpoznávání vytvořené v jazyce C++. Po skončení třídění mincí se všechny počty hodnot jednotlivých mincí ukládají do Archiveru. V dřívější verzi byl jeden archiver, do kterého se ukládaly počty mincí všech měn, i když byla zrovna tříděna jen jedna. Nyní jsem zvolila pro každou měnu jeden archiver pro lepší přehlednost archivního souboru. Pak už jsem provedla jen drobné grafické změny, jako přesunutí Start tlačítka a počítání doby strávené tříděním či změnu pozadí celé aplikace. Náhled aplikace s výběrem třídění jedné mince můžete vidět na obrázku 29. Vzhled s variantou pro více měn můžete vidět v příloze na obrázku 47



Obrázek 29: Aplikace v Control webu s výběrem jedné měny.

4.13 Výpočet počtu kroků motoru

Vzhledem k tomu, že v mé práci třídím 32 různých měn, bylo potřeba také spočítat počty kroků pro každou minci ve všech měnách. Jak už bylo popsáno výše v kapitole 4.10, u měn s počtem 7 a více různých hodnot mincí jsou umísťovány 2 různé hodnoty do dvou krabiček. Na druhou stranu u měn s 5 a méně různých hodnot mincí v měně je počet kroků na každou minci stejný a jen se zmenšuje počet využitých krabiček. Zvolila jsem tedy jen 2 různé počty kroků pro zařídování.

Jeden způsob je po 4, 8, 12, 16 a 21 krocích krokového motorku, který využívají měny SEK, KRW, MYR, NOK, RON, AUD, INR, IDR, NZD a SGD. Zde například u měny SEK budou využity jen 4 krabičky a to pro 3 různé hodnoty mincí dané měny a 1 krabička pro nerozpoznané mince. U měn AUD, INR, IDR, NZD, SGD budou využity všechny krabičky.

Druhý způsob je po 3, 7, 11, 14, 18 a 21 krocích krokového motorku. Tímto způsobem se třídí ostatní měny. U měn BGN, PHP, HKD, CAD a CHF se umísťují jen první dvě různé hodnoty mince do jedné krabičky, dále u měn GBP, EUR, MXN jsou do dvou krabiček umísťovány 4 různé hodnoty mincí, u PLN a THB do tří krabiček 6 různých hodnot mincí a v uměny HRK je takto do 4 krabiček umístěno 8 druhů mincí. Tento problém s tříděním více druhů mincí do jedné krabičky by vyřešilo rozšíření modelu, jak je popsáno v kapitole 4.10.

4.14 Moduly pro ovládání prvků

Pro ovládání elektronických prvků mi slouží jednotka průmyslových vstupů a výstupů s USB rozhraním DataLab IO. Tato jednotka, aby fungovala, musí se doplnit potřebnými moduly. Pro svoji práci jsem potřebovala 3 různé moduly. Jeden vstupní a dva výstupní. Vstupní modul je využíván pro zjišťování signálu, kdy jsou spínací tlačítka sepnutá. K tomu jsem použila modul DI1H, modul nezávislých digitálních vstupů. Logická jednička na vstupu je indikována LED na modulu. Označení H V je na vstupu logická nula (false), při napětí 8 až 35 V je na vstupu logická jedna (true). Pro spínání relé a osvětlení byl zvolen modul DO1, modul réleových vstupů. Ten obsahuje 8 réleových spínacích kontaktů. LED osvětlení na modulu se rozsvítí při sepnutém výstupu. Jako poslední modul pro ovládání krokového motoru byl vybrán modul DO2, modul digitálních vstupů s otevřeným kolektorem. Ten obsahuje 8 digitálních spínacích vstupů s otevřeným kolektorem, kde jsou všechny výstupy navzájem galvanicky oddělené. Zde je nutné dodržet polaritu výstupního signálu, jelikož jinak ochranná ioda drží výstup trvale sepnutý. Proto na svorku každé dvojice s označením .1 musí být vyveden emitor spínacího tranzistoru a na svorku .2 kolektor spínacího tranzistoru.

4.15 Rozpoznávání obrazu

Pro rozpoznávání obrazu jsem zvolila 4 různé metody. První dvě vybrané metody, SIFT a SURF byly vybrány, jelikož vykazují dobré výsledky u rozpoznávání obrazu a jejich implementace není náročná. Bylo trochu jasné, že pro rozpoznávání mincí tyto metody asi nebudou vhodné, ale chtěla je vyzkoušet a popsat jejich funkčnost na daném problému. Další dvě metody byly vybrány ze skupiny umělých neuronových sítí. Hopfieldova síť byla vybrána pro dobré výsledky pro rozpoznávání textu a menšího množství obrazu. Vícevrstvá neuronová síť byla už trochu jistota pro rozpoznávání obrazu, pokud se mi podaří vzory před učením dobře předzpracovat.

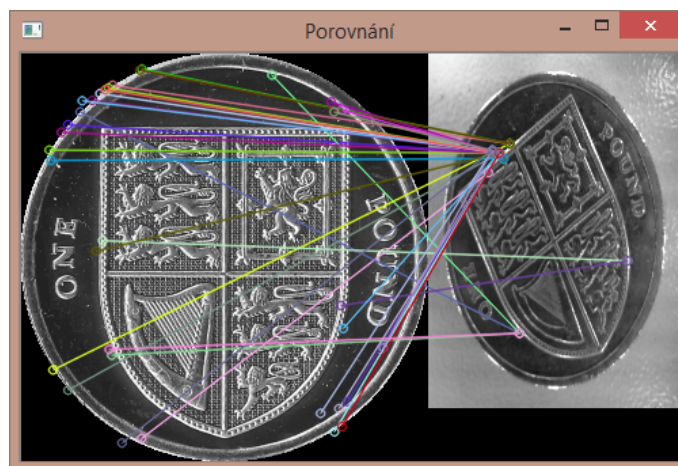
4.16 Předzpracování obrázu pro metody SURF a SIFT

Pro využití jakékoli metody rozpoznávání je nutno nejdříve upravit daný vstup. V případě využití metod SIFT a SURF pro rozpoznávání obrazu je nutné upravit vstupní obrazy. V prvním kroku jsem načetla vstupní obraz v odstínech šedi a uložila do proměnné typu Mat. Načtení v odstínech šedi jsem zvolila z toho důvodu, aby nebyl problém u rozpoznávání s různými odstíny barev. Ten by se mohl objevit u jakékoli malé odchylky osvětlení s menší odlišností barvy. Do matice Mat se uložily hodnoty pixelů od 0 do 255, kde 0 je bílá a 255 je černá. Všechny vstupní obrazy mají stejný rozměr. Matice Mat zachovává stejný rozměr jako je vstupní obraz. Například pokud má vstupní obraz 300×300 pixelů, i tato matice má rozměr 300×300 . Jelikož se může stát, že adresa vstupního obrazu je špatně zadaná, je potřeba kontrolovat, zda byl vstup správně načten. V případě nenačtení vstupního obrazu je další proces zpracování a rozpoznávání zbytečný. Proto se proces pro tento obraz ukončí a přejde se na další vstupní obraz. Takto zpracovaný a uložený vstup lze dále využívat pro rozpoznávání obrazu.

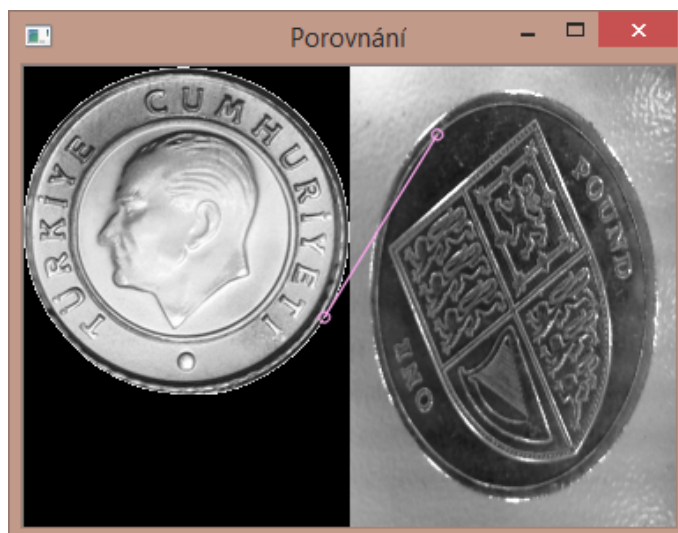
4.17 FLANN knihovna s metodou SURF

První z využitých metod rozpoznávání obrazu v této práci byla FLANN knihovna s použitím metody SURF pro nalezení klíčových bodů a následně pro výpočet deskriptorů. Jak bylo již dříve popsáno, tato metoda je založená na rychlém hledání nejbližších sousedů ve vysoce dimenzionálních prostorech. Metoda je naprogramována tak, že první krok algoritmu je vyhledávání klíčových bodů pomocí SURF detektoru. Po nalezení všech klíčových bodů se k těmto bodům vypočtou deskriptory opět pomocí SURF metody. To vše se musí provést pro oba obrazy, které chceme porovnat. Ve třetím kroku se porovnávají vektory deskriptorů pomocí knihovny FLANN, která porovná všechny nalezené vektory deskriptorů z prvního a druhého obrazu. Nakonec ještě tato metoda projde všemi klíčovými body a uloží si jen ty klíčové body, které jsou do velikosti dvojnásobku minimální vzdálenosti mezi nimi či jejich hodnota je maximálně 0,2. Výsledkem je pak počet přiřazených klíčových bodů z obou obrazů. Pro tuto práci stačí jen počet těchto bodů. Tyto body i s obrazy si můžeme zobrazit pro kontrolu, které body v obraze byly použity.

Po naprogramování této metody jsem otestovala, zda by bylo dobré metodu použít pro rozpoznávání mincí. Výsledkem tohoto testu bylo porovnávání mnoha klíčových bodů s jinými částmi mincí a nakonec jsem došla k závěru, že tato metoda je nepoužitelná. Například u nafocené anglické mince 1 Libra, bylo při porovnání se vzorem 1 Libry jen 33 společných klíčových bodů. Metoda přiřadila tuto minci k minci amerického dolaru, a to 1 centu s 986 společnými klíčovými body. Nejméně společných klíčových bodů (1 klíčový bod) bylo s 50 kurusy turecké liry.

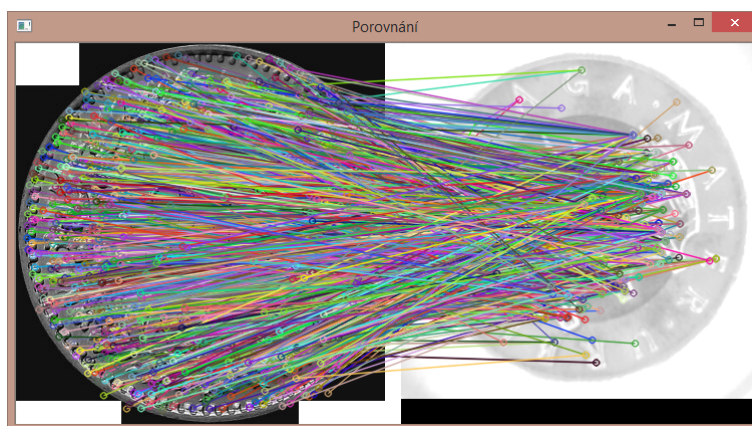


Obrázek 30: Metoda FLANN s SURF - Výsledek porovnání nafocené 1 Libry se svým vzorem.

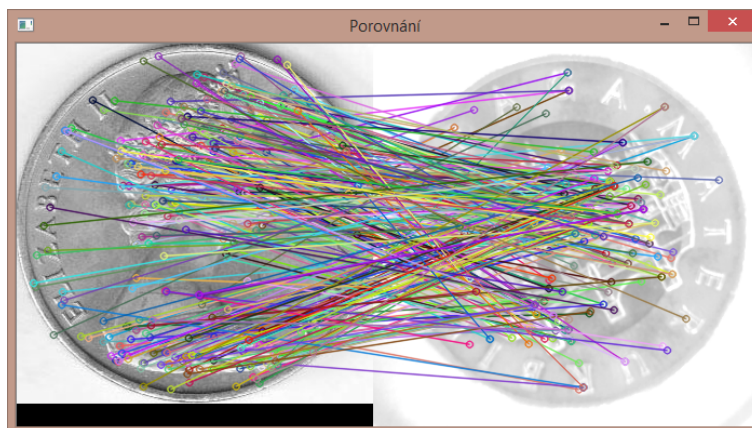


Obrázek 31: Metoda FLANN s SURF - Výsledek porovnání nafocené 1 Libry se vzorem 50 kurusů turecké liry.

Druhým porovnáním se testovala mince, která byla vícebarevná. Sice se v této metodě používá porovnání obrazů v odstínech šedi, ale i tak by to mohlo porovnání pomoci. Nakonec i při tomto porovnání nebyla správně přiřazena nafocená mince se svým vzorem. Testovala jsem naši českou padesátikorunu a nejvíce společných deskriptorů měla se švýcarskou mincí 20 centů, a to přesně 1115 deskriptorů. Padesátikoruna se svým vzorem měla společných pouze 50 klíčových bodů, což je hodně málo. Pouze 9 společných klíčových bodů bylo při porovnání 50 koruny s 10 centy australského dolaru.



Obrázek 32: Metoda FLANN s SURF - Výsledek porovnání nafocené 50 koruny se vzorem 20 švýcarských centů.



Obrázek 33: Metoda FLANN s SURF - Výsledek porovnání nafocené 50 koruny se vzorem 10 australských centů.

Pro hlubší analýzu porovnávání mincí touto metodou jsem naprogramovala test, kdy se pro každý vzor vypočetlo pořadí podobnosti, aby se zjistilo, kolik mincí mělo více klíčových bodů než nafocená mince se svým vzorem. Nafocené mince pro testování byly české koruny, anglické libry, maďarské forinty, chorvatské kuny a americké dolary.

Nejhůře v těchto testech dopadly české koruny. Z nich byla na tom nejlépe mince o hodnotě 1 koruny, která byla nafocena z aversní strany. I tak byl vzor přiřazen k fotografii až jako 169. mince. U českých korun se našlo i porovnání, kde nebyl nalezen ani jeden klíčový bod. Stalo se tak u reversní strany mince 5 Kč.

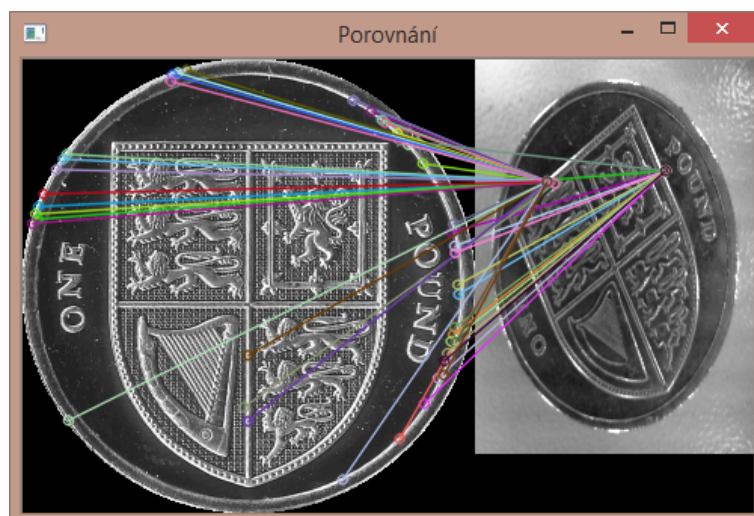
O poznání lépe v těchto testech uspěly nafocené mince anglické libry a amerického dolaru. U liber byla dokonce 3. v pořadí mince o hodnotě 5 pencí z reversní strany. U amerických dolarů bylo toto pořadí ještě lepší a minci o hodnotě 5 centů bylo přiřazeno druhé pořadí.

Jak lze vidět z výsledků této hlubší analýzy pro SURF metodu rozpoznávání, žádná mince by nebyla přiřazena správně. Závěrem lze tedy konstatovat, že tato metoda není pro porovnávání mincí vhodná.

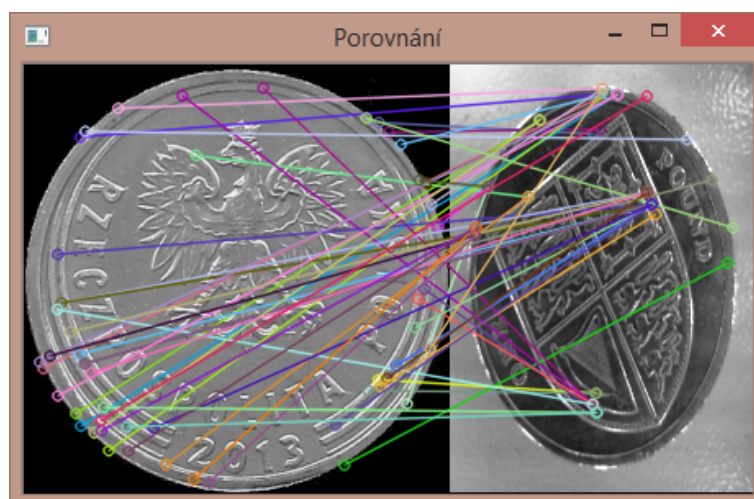
4.18 FLANN knihovna s metodou SIFT

Druhá vybraná metoda pro porovnávání mincí byla také knihovna FLANN, ale tentokrát s využitím SIFT metody pro vyhledávání klíčových bodů a následném výpočtu deskriptorů. Jako u dřívější metody se v prvním kroku vyhledávaly klíčové body, ze kterých se ve druhém kroku vypočítaly deskriptory. Poté pomocí FLANN knihovny se porovnály vektory deskriptorů a následoval výpočet maximálních a minimálních vzdáleností mezi klíčovými body. V posledním kroku se vybraly jen ty klíčové body, které měly vzdálenost menší než dvojnásobek minimální vzdálenosti klíčových či hodnotu menší než 0,02. Nakonec jsem mohla oba porovnávané obrazy zobrazit a zobrazit i jejich společné klíčové body.

Pro první otestování jsem stejně jako u předchozí metody vybrala anglickou libru o hodnotě 1 Libra a českých 50 Kč. Anglická libra měla nejvíce společných klíčových bodů s mincí polské měny (2 grosze), a to 55 klíčových bodů. V případě porovnání se vzorem anglické libry měly společných jen 44 klíčových bodů. Hned u této mince je vidět, že tato metoda není použitelná pro porovnávání těchto obrazů. Vůbec žádný společný klíčový bod nebyl nalezen porovnáním s českou mincí 5 Kč.

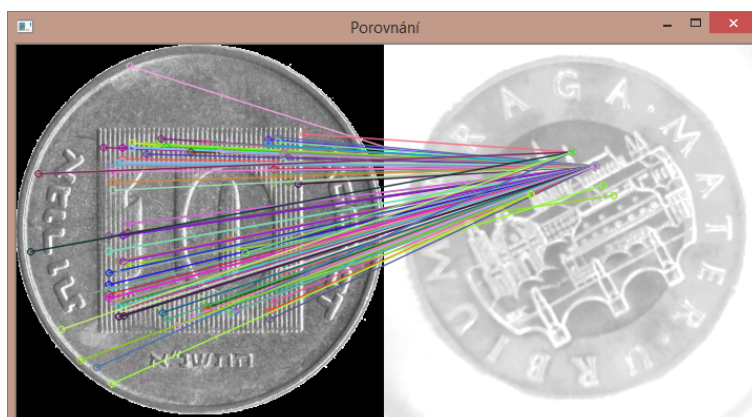


Obrázek 34: Metoda FLANN se SIFT - Výsledek porovnání nafocené 1 Libry se svým vzorem.

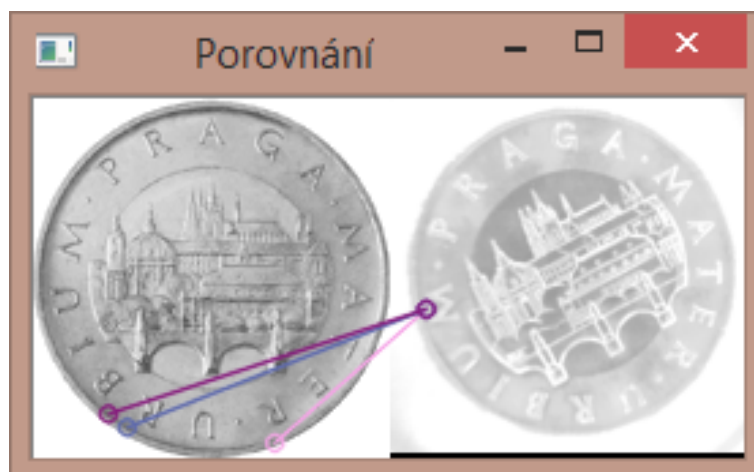


Obrázek 35: Metoda FLANN se SIFT - Výsledek porovnání nafocené 1 Libry se vzorem polských 2 groszů.

Ještě hůře dopadlo porovnání obrazu české 50 Kč. Zde se svým vzorem měl obraz jen 3 společné body. Tato metoda by přiřadila k nafocené 50 Kč minci izraelského šekelu (10 agorot), a to se 121 společnými klíčovými body. I zde by tedy přiřazení neproběhlo správně.



Obrázek 36: Metoda FLANN se SIFT - Výsledek porovnání nafocené 50 koruny se vzorem 10 agorot izraelského šekelu.



Obrázek 37: Metoda FLANN se SIFT - Výsledek porovnání nafocené 50 koruny se vzorem 50 koruny.

4.18.1 Závěr k metodě SIFT

I na tuto metodu jsem provedla hlubší analýzu porovnávání, zda se některá nafocená mince přiřadí ke svému vzoru. Analýza byla spuštěna na stejných vyfocených mincích jako v metodě SURF, tedy na mincích měn české koruny, maďarského forintu, chorvatských kun, amerických dolarů a anglických liber. Výsledek vedl k tomu, že tato metoda porovnávání je ještě horší než metoda předchozí, s využitím SURF. Opět nejhorší bylo porovnávání mincí české měny. Zde ve více případech nebyl nalezen ani jeden společný klíčový bod se svým vzorem. Maximum klíčových bodů u mincí české měny v porovnání se svým vzorem bylo jen 24, což vedlo k tomu, že by tato mince byla ke svému vzoru přiřazena až jako 277 v pořadí. Nejlepší porovnávání proběhlo u nafocených mincí anglické libry a amerických dolarů. Ovšem u mincí anglické libry se stále počet společných klíčových bodů většinou pohyboval

kolem 50. U 7 nafocených mincí se počet společných klíčových bodů přehoupl přes 100. U amerických dolarů počet společných bodů byl většinou vyšší než u ostatních měn, nejlépe na tom byla mince s 301 společnými klíčovými body, ale ke svému vzoru by byla přiřazena jako 10. Z těchto výsledků je patrné, že ani tato metoda se nedá využít pro třídění mincí, jak už pro vzory mincí, které jsou z ČNB anebo mnou nafocených.

4.19 Hopfieldova síť

4.19.1 Vstupní data

Ve vstupních datech mám všechny mince, které má Hopfieldova síť rozpoznat. Vstupem je tedy celkem 400 vzorů, a to mince jak z aversní strany mince, tak i z reversní strany. Vstupní množina mincí, které jsou pro tuto práci vybrané, aby byly rozpoznávány, mají své adresy umístění uloženy v souboru, z kterého se po řádcích načítají. Na každém řádku je jedna mince.

4.19.2 Předzpracování obrazu pro Hopfieldovou síť

Každý obraz se musel prvně předzpracovat. To znamenalo změnit jeho rozměr, aby všechny vstupní data měly stejný rozměr a nedošlo k problému u rozpoznávání kvůli různým rozměrům vstupů. Dalším krokem pro předzpracování bylo změna obrazu z RGB modelu na odstíny šedi. Jelikož Hopfieldova síť pracuje jen s binárním nebo bipolárním vstupem, bylo potřeba dále tento vstup upravit. Na toto jsem využila prahování, kde jsem si zvolila hodnotu prahu pomocí postupného testování. Hodnotu prahu jsem zvolila 200. Hodnotám pixelů menších než 200 jsem přiřadila hodnotu -1 (bílá) a hodnotám pixelů větších než 200 hodnotu 1 (černá). Mince jsou většinou světlejší, takže pokud by byla hodnota prahu v polovině rozsahu odstínu šedi, tedy na 128, byl by vstup z větší části bílý a nebylo by možno rozpoznávat.

Takto upravené hodnoty vstupu jsem uložila do vektoru. Jelikož jsem potřebovala u každého vstupu zaznamenat i jeho název, musela jsem využít typ proměnné Map, kde jako klíč je název mince a hodnota je vektor hodnot daného vstupního obrazu.

Tímto způsobem jsem zpracovala všechny vstupní obrazy, které sloužily jako vzory do trénovací množiny pro učení Hopfieldovy sítě.

4.19.3 Trénovací množina

Do trénovací množiny, jak je popsáno výše, byly zpracované vstupní obrazy uloženy do proměnné typu Map. Z takto zpracované trénovací množiny už lze vypočítat potřebnou váhu každého neuronu. Algoritmus pro počítání vah neuronů je níže:

```
for (int i = 0; i < sloupcu; i++){
    for (int j = 0; j < sloupcu; j++){
        for (map<string, vector<int>>::iterator it =
            poleHopfield.begin(); it != poleHopfield.end(); it++){
            if (i != j){
                pom = (int)it->second[i] * (int)it->second[j];
                w[i][j] += pom;
            }
        }
    }
}
```

Váhy se ukládají do proměnné w typu matice. Velikost této matice se odvíjí od velikosti vstupních obrazů. Například pro velikost obrazu 30×30 bude velikost této matice 900×900 . V případě, že by hodnota iterátorů i a j byly stejné, hodnota této váhy je 0. V jiném případě se vynásobí hodnoty vstupních pixelů a uloží se na dané místo proměnné.

4.19.4 Rozhodování

Proces rozhodování u Hopfieldovy sítě je následující. Z vypočtené váhy trénovací množiny a vstupního obrazu se spočtou hodnoty u , které jsou potřebné pro další pokračování výpočtu rozhodování. Hodnoty u jsou pro každý sloupec jiné a vypočítají se přičítáním každé váhy daného sloupce vynásobené hodnotou vstupního obrazu pro rozhodování, také z daného sloupce.

```
for (int j = 0; j < sloupcu; j++){
    for (int i = 0; i < sloupcu; i++){
        u[j] += w[j][i] * k[i];
    }
}
```

Po kompletním výpočtu u se přejde k transformační funkci, která rozhoduje o tom, jaké hodnoty bude mít hodnota vektoru k v následujícím kroce $k+1$. Pak se porovnává každá hodnota u , zda je kladná, záporná či rovná nule. V případě kladné hodnoty se do odpovídajícího $k+1$ vloží hodnota 1, v případě záporné hodnoty se vloží hodnota -1 a v případě hodnoty rovnající se nule, se vloží nynější odpovídající hodnota k .

```
for (int j = 0; j < sloupcu; j++){
    if (u[j] > 0){
        kPlus1[j] = 1;
    }
    else if (u[j] < 0){
        kPlus1[j] = -1;
    }
    else if (u[j] == 0){
        kPlus1[j] = k[j];
    }
}
```

Nyní už následuje jen porovnání, zda celé pole hodnot $k+1$ je rovno celému poli hodnot k . V případě nerovnosti se začíná opět od začátku rozhodování, tedy s výpočtem hodnot u , přes transformační funkci až po porovnávání těchto dvou polí. Maximální počet opakování jsem nastavila na 500. Pokud ani po 500 opakováních se tyto dvě pole nerovnejí, je jasné, že Hopfieldova síť nedošla k žádnému správnému výsledku a obrazy se nerovnejí.

To však neznamená, že při dřívějším ukončení, kdy se pole rovnají, byl nalezen nějaký vzor. Může se stát, že se Hopfieldova síť nepřiklonila ani k jednomu vzoru a vytvořila neznámý artefakt. To zjistíme jednoduše, když se vypočítaná hodnota k porovná se vzory, které byly předloženy jako trénovací množina. Nalezením nějakého stejného vzoru proces skončil a vypíše se, který vzor odpovídá minci vložené do rozpoznávání. V případě nenalezení vzoru se vypíše vzor nenalezen.

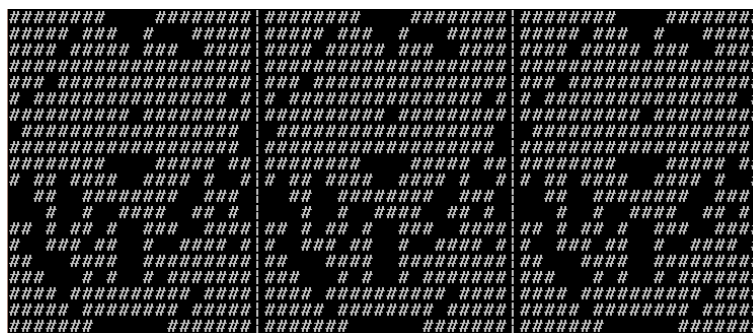
4.19.5 Natočení vstupních dat pro rozhodování

Při focení mince v modelu nelze zajistit, aby každá mince byla stejně natočená. Například, aby každá mince měla základní natočení 0° . Jelikož Hopfieldova síť neumí rozeznat obraz, která je jinak natočený, musela jsem tento problém vyřešit ručně. Natočila jsem každou minci do trénovací množiny tak, aby byla co nejmenší pravděpodobnost nerozpoznání. Zvolila jsem úhel natočení 10° , a tím mi vzniklo pro každou minci 36 vzorů. Tím se ale prodloužila doba učení Hopfieldovy sítě. Stále však nebylo zaručeno správné rozpoznání, jelikož mohla být mince na rozpoznání vyfocena právě mezi těmito úhly, a tak by její Hopfieldova síť nemusela rozpoznat. To jsem zkusila řešit natáčením rozpoznávaného obrazu o jeden stupeň a porovnávat s danými vzory. Bohužel tato cesta se neukázala správná, vzory byly hodně podobné, a proto bylo velké množství mincí nerozpoznáno. Toto je problém Hopfieldovy sítě, kdy při větší trénovací množině podobných obrazů Hopfieldova síť není schopna rozpoznávat.

4.19.6 Určení hodnoty mince

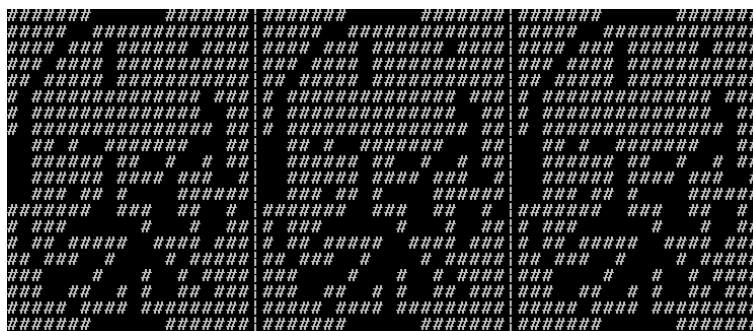
Ani tato metoda pro rozpoznávání mincí nefunguje. Ve zdrojích tohoto učení byly uváděny příklady s rozpoznáváním textu, ale ne takového obrazu, jako mám já pro diplomovou práci.

Když jsem testovala rozpoznávání úplně rozdílných mincí, jako například 1 Kč a 10 Kč ze reversní strany, nebyl problém. To lze vidět na obrázku 38, kde jako první dva ASCII obrázky jsou z hodnoty pixelů výpočtu Hopfieldovy sítě v posledním kroku. Třetí ASCII obrázek je správný obrázek, tedy obraz z trénovacího vzoru 1 Kč z reversní strany, ke kterému by se měla Hopfieldova síť dopočítat, aby mohla prohlásit, že jde o tuto minci.



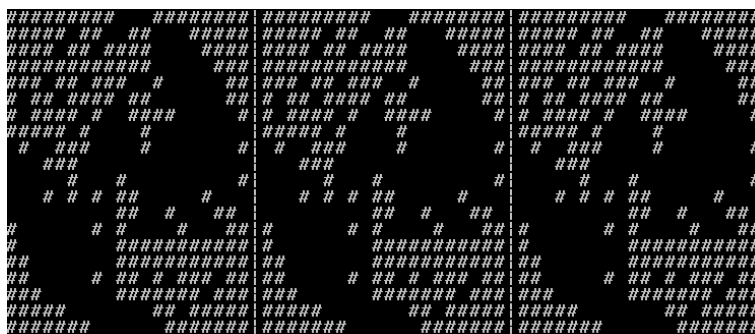
Obrázek 38: Úspěšné rozpoznání mince o hodnotě 1 Kč z reversní strany.

Na dalším obrázku je vidět výsledný výstup výpočtu Hopfieldovy sítě pro testovací minci o hodnotě 10 Kč z reversní strany. Toto rozpoznání také proběhlo v pořádku.



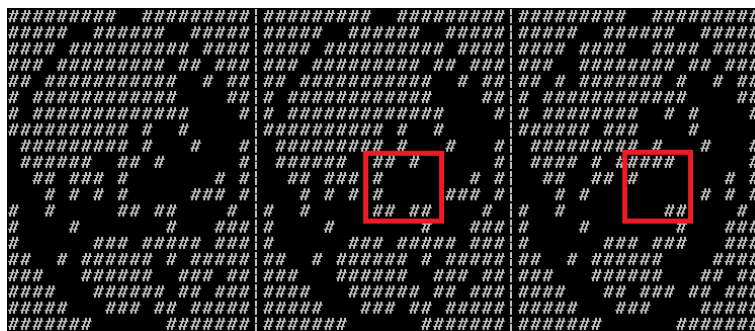
Obrázek 39: Úspěšné rozpoznání mince o hodnotě 10 Kč z reversní strany.

Ale postupně, když jsem přidávala do trénovací množiny více vzorů, byl problém rozpoznání čím dál větší. Pokud jsem přidala pouze jednu minci do trénovací množiny, minci o hodnotě 20 Kč s reversní stranou, rozpoznávání bylo v pořádku a Hopfieldova síť došla ke správnému cíli.



Obrázek 40: Úspěšné rozpoznání mince o hodnotě 20 Kč z reversní strany.

Při přidávání dalších mincí do trénovací množiny s reversní stranou mince se však nerozpoznávaly nejen tyto mince, ale Hopfieldova síť přestala správně rozpoznávat i některé mince, které s menší trénovací množinou rozpoznala. Na obrázku 41 můžeme vidět vyznačený kus ASCII obrazu, který by se měl shodovat, aby byl vypočtený výstup prohlášen za rozpoznáný, jenže se tyto vyznačené kusy obrazu liší a rozpoznání neproběhlo správně.



Obrázek 41: Neúspěšné rozpoznání mince o hodnotě 50 Kč z reversní strany.

Proběhlo pár testování i na mince měny Euro, ale i zde byly výsledky špatné. Správné byly jen při dvou, maximálně třech mincích v trénovací množině. Více měn jsem u této metody rozpoznávání ani nezkoušela, jelikož bylo jasné, že tato metoda nebude dobrá. A když jsem zkoušela porovnávat aversní strany mincí, které jsou u většiny mincí stejné, bylo správné rozpoznání mincí s minimální. Úspěšné bylo jen v případě, že dvě mince v trénovací množině měly úplně jinou aversní stranu. Například mince měny Euro a mince české koruny. Ale rozpoznání neproběhlo úspěšně u všech mincí a jakmile bylo v trénovací množině mincí více, problém rozpoznávání se zvýšil.

Dalším problémem, který tato metoda neumí řešit, je rozpoznání natočené mince. V případě vložení do trénovací množiny více kusů stejných mincí jen jinak natočených nebyl úspěch rozpoznávání velký. Správně rozpoznaná mince v nějakém natočení byla v maximálně 25 %, a to v trénovací množině nesměla být mince o jiné hodnoty.

4.19.7 Závěr Hopfieldovy sítě

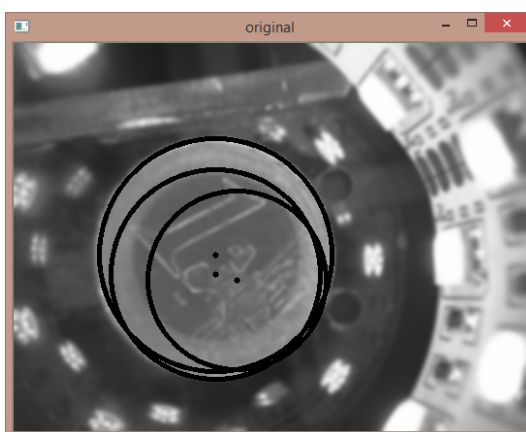
Po testování Hopfieldovy sítě musím konstatovat, že se tato síť pro rozpoznávání mincí nehodí. Velké množství mincí je podobných a potom Hopfieldova síť neví, ke kterému vzoru se má přiklonit. Další problém Hopfieldovy sítě byl ten, že nebyla schopna porovnat mince jinak natočené a muselo se to řešit dalším přidáváním vzorů do tréninkové množiny. Tím se ještě zvětšil problém velké trénovací množiny s podobnými vstupy. Chybovost rozpoznávání Hopfieldovy sítě byla u mincí velká.

4.20 Vícevrstvá neuronová síť

Poslední zvolenou metodou pro třídění mincí byla vícevrstvá neuronová síť. Tato neuronová síť pracuje s velkým množstvím vstupů, a proto jsem zvolila učení neuronové sítě v programu Matlab. Učení probíhalo pomocí algoritmu backpropagation. Mohla bych tento algoritmus naprogramovat, ale jelikož v Matlabu je tento učící se algoritmus optimalizován, zvolila jsem učení v Matlabu. Rozhodování vícevrstvé neuronové sítě už jsem pak naprogramovala v jazyce C++.

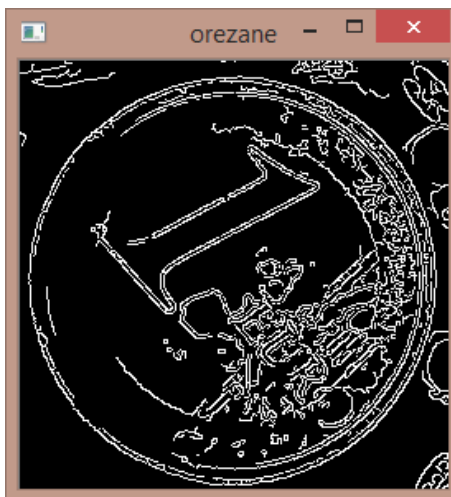
4.20.1 Načtení vstupních obrazů

Pro práci s vícevrstvou neuronovou sítí je potřeba mít velké množství vstupních obrazů pro trénovací množinu. Zpracovala jsem 400 vstupních obrazů v odstínech šedi pro každou minci. Každý vstup jsem si načetla do proměnné typu Mat. Pak proběhla kontrola načtení vstupu. Jelikož jsem se rozhodla pro měření rozměru přímo v aplikaci rozpoznávání, musela jsem naprogramovat nalezení kruhu ve vstupním obraze. I když jsem zadala minimální hledaný rozměr, aplikace někdy našla více kruhů než jen jeden požadovaný, viz obrázek 42. Hledaný rozměr jsem vybrala zvolením největšího nalezeného kruhu v obraze.



Obrázek 42: Vstupní obraz v odstínech šedi a s nalezenými kruhy.

Když už jsem naprogramovala nalezení požadovaného kruhu, využila jsem jej pro další funkce předzpracování obrazu. Abych nemusela učit neuronovou síť s velkým obrazem, kde mince zabírá jen menší část, pomocí kruhu jsem tento vstupní obraz ořízla. Na osu x a y od středu kruhu jsem na obě strany přidala 10 pixelů a podle tohoto rozměru jsem obraz ořízla do tvaru čtverce. Na tento oříznutý obraz jsem aplikovala Cannyho detektor hran pro získání binárních hodnot obrazu.



Obrázek 43: Vstupní obraz oříznutý a v binárních hodnotách.

Další funkce, pro kterou byl využit nalezený kruh bylo odstranění okolí mince. Všem pixelům, které jsou okolo mince, byla přiřazena hodnota 255, což je hodnota bílé barvy.



Obrázek 44: Vstupní obraz s vymazaným pozadím.

Takto upravenému obrazu jsem změnila rozměr na velikost 50×50 pixelů. Tato změna velikosti se nemohla provést dříve než byla mince změřena v aplikaci, jelikož by pak podle jednotného rozměru se nedal určit druh mince. Kdyby byl jednotný rozměr ještě menší, už by nebylo možné mince vůbec rozpoznávat. Ale i naopak, čím větší vstup, tím delší by bylo zpracování a rozpoznávání. Hodnoty obrazu byly uloženy do vektoru a pak uloženy do souboru. Za těmito trénovacími vzory musely být ještě zpracovány vzorky obsahující všechny ostatní mince. Pro každou takovou minci stačily 2 vzory pro jednu stranu mince. Tyto vzorky byly využity, aby se umělá neuronová síť nenaučila jen na jednu minci a vše ostatní by také označila za tuto minci. Počet 2 vzorů pro jednu stranu mince při testování měl správné výsledky pro přibližně 800 vzorů mince, pro kterou je neuronová síť určena. V případě většího počtu vzorů jedné mince je potřeba větší počet vzorů ostatních mincí. Na konec této trénovací množiny jsem vložila ještě vzorek o hodnotách 0 a vzorek o hodnotách 1. To proto, že při trénování vícevrstvé neuronové sítě v Matlabu se pro rychlejší učení vstup ořízne, pokud se na konci všech vzorů vyskytují stejné hodnoty. Vložením těchto dvou vzorků na konec souboru zamezím ořezání vstupních dat.

Při zapisování vstupních vektorů do souboru se vkládaly do dalšího souboru data, která udávají, jaký je očekávaný výstup z neuronové sítě pro daný vstup. Tato data mají pro každý vzor 6 binárních hodnot ve tvaru například 1 0 1 0 1 0. Očekávaný výstup pro vzory ostatních mincí, které jsou v trénovací množině, jsem zvolila samé nuly (0 0 0 0 0 0). Na konci souboru jsou opět vloženy dva vzorky s hodnotami 0 a 1.

Obraz bylo nutno pro rozpoznávání ještě zpracovat. Adresa obrazu pro rozpoznávání je získána z parametru, který předává Control Web. Zpracování tohoto obrazu se provádělo naprosto stejně jako obrazy pro trénovací sadu, jen s jediným rozdílem, kdy bylo potřeba na první index vektoru přidat konstantu před vektor hodnot obrazu. Tuto konstantu jsem neukládala do tohoto vektoru, ale řešila jsem ji přidáním až přímo v části rozhodování.

4.20.2 Učení neuronové sítě pomocí Matlabu

Pro učení vícevrstvé neuronové sítě jsem zvolila program Matlab, kde učení probíhá pomocí algoritmu backpropagation. Pro toto učení jsem potřebovala vstup obrázku v binární podobě. Vstup musí být formátován tak, že každý sloupec znamená i-tý prvek daného obrázku. Například pro 2. prvek 3. obrázku je potřeba mít hodnotu z 3. řádku a 2. sloupce. Takto upravené musí být data pro vstup a rovněž data pro cíl, tedy jak má vypadat výstup daného vstupního vzoru. Jakmile jsem měla data připravena, spustila jsem příkazem *nnstart* v Matlabu Neural Network Toolbox pro práci s umělými neuronovými sítěmi. Mohla jsem vybírat z jejich několika možností, pro rozpoznávání vzoru je možnost *Pattern recognition and classification*. V dalším kroku se vloží dané vstupní hodnoty input a cílové hodnoty target. Důležité nastavit, jak formátování dat. Pro moje soubory je nutné zvolit položku Matrix rows. Po úspěšném načtení těchto dat si uživatel volí, kolik procent dat je pro trénování, ko-

lik pro testování a kolik pro validaci. V neposlední řadě je nutno nastavit potřebný počet neuronů ve skryté vrstvě, já jsem zvolila hodnotu 400. Následující krok už spouští učení neuronové sítě, které probíhá pomocí algoritmu backpropagation. Po naučení a otestování neuronové sítě se ukáže, kolik času (time) trénování potřebovalo, kolik cyklů (epoch) proběhlo, performance, gradient, validation checks. Dále je možné se podívat na různé grafy jako postup performance, training state, histogram chyby (error histogram), confusion. Pro mou práci bylo potřeba získat z této aplikace hodnoty vah po učení. Aplikace automaticky nabízí jen výsledky z testování, váhy se musí získat jiným způsobem. Z toolboxu jsem si musela uložit všechny výsledky i síť do workspace projektu. Potom jsem pomocí příkazu *getwb(net)* uložila do proměnné *w* hodnoty všech vah. Ty jsem z Matlabu uložila do csv souboru příkazem *dlmwrite('navez-vahy.csv', w, 'delimiter', ',', 'precision', 15)*; První parametr funkce je název váhy, druhý parametr je proměnná, v které jsou váhy uloženy, pak následuje určení oddělovače hodnot a nakonec přesnost, na kolik desetinných míst se má hodnota zaokrouhlit. Pro uložení dat do csv existuje v Matlabu také příkaz *csvwrite()*, ten ale není vhodný, jelikož zaokrouhluje hodnoty na 4 desetinná místa. Pro rozhodování ve vícevrstvé neuronové síti je nutné mít co nejpřesnější hodnoty. Z aplikace lze zjistit, která data na vstupu byla použita jako trénovací, validační a testovací. Toto se dá využít při testování neuronové sítě. Pro otestování správné funkčnosti vícevrstvé neuronové sítě je potřeba použít obrazy, které nebyly v testovací množině.

Nyní na ukázkou skriptu, jak jsem trénovala a testovala vícevrstvou neuronovou síť v Matlabu:

```
inputs = input_data;
targets = target_data;

% Vytvoreni vicevrstve neuronove site
hiddenLayerSize = 400;
net = patternnet(hiddenLayerSize);

% Nastaveni procenta vyuziti dat pro trenovani, validaci a
testovani
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% trenovani site
[net, tr] = train(net, inputs, targets);

% testovani site
outputs = net(inputs);
errors = gsubtract(targets, outputs);
performance = perform(net, targets, outputs)
```

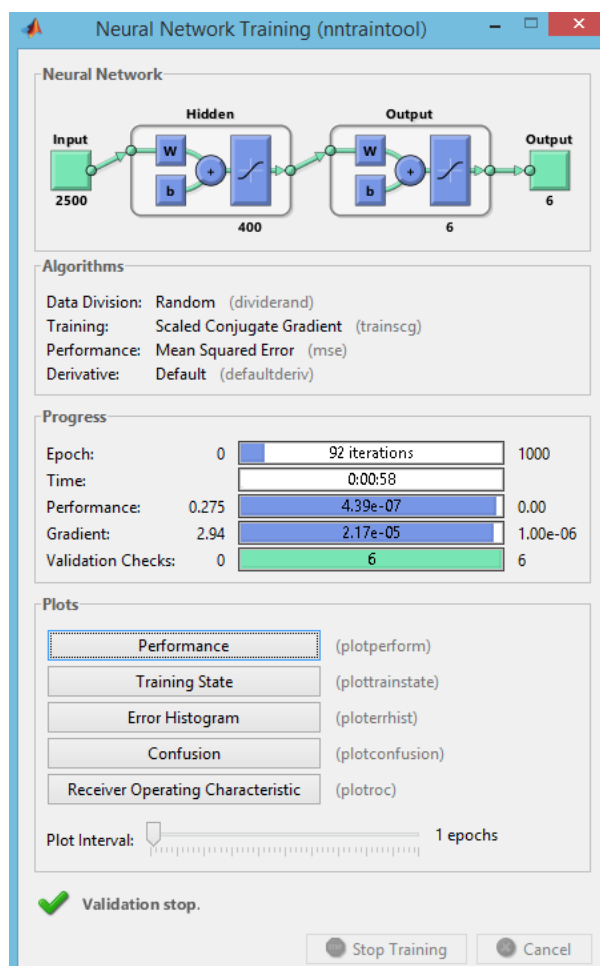
```

\newpage
% Zobrazení sítě
view(net)

% uložení vah do proměnné a uložení do csv
w = getwb(net);
csvwrite('vahy.csv',w)

```

Pro rychlejší rozpoznávání mincí jsem zvolila naučit tolik neuronových sítí, kolik mám mincí celkem. To znamená, že každá neuronová síť bude umět rozpoznat jednu minci. Pro rozpoznávání se vždy použijí ty naučené neuronové sítě, které odpovídají danému rozměru mince. Nebude tedy potřeba rozpoznávat minci s naučenou neuronovou sítí pro 197 mincí, ale použije se například jen 10 neuronových sítí. To běh aplikace mnohonásobně zrychlí.



Obrázek 45: Výsledky učení neuronové sítě.

4.20.3 Načtení vah do proměnných

Po naučení vícevrstvé neuronové sítě v programu Matlab a uložení vah jednotlivých vrtev sítě do souboru bylo potřeba tento soubor zpracovat. Po zhlédnutí obsahu souboru jsem zjistila, že každá váha je oddělena řádkem. Z toho důvodu nebylo potřeba řešit mezery mezi hodnotami či případně středníky, i když se jedná o soubor formátu csv. Dále bylo potřeba zjistit, jak jsou tyto váhy uspořádány. Potřebovala jsem najít matici vah pro skrytou vrstvu o rozměru 2501×400 . 2501 je počet pixelů vstupu ke kterému je přidána na začátek konstanta. 400 je počet neuronů ve skryté vrstvě. Bylo nutné zjistit, jak váhy správně načíst, aby se správně uložily do potřebné matice. Jelikož jsem věděla, že první hodnota v každém řádku je konstanta, soustředila jsem se na hledání těchto konstant v souboru. Ty byly na začátku souboru jako prvních 400 hodnot a tak bylo zřejmé, jak hodnoty ze souboru načíst. Hodnoty vah načítat a poté ukládat postupně po sloupcích. Po zpracování těchto vah byly v souboru ještě další hodnota. To byly hodnoty vah pro výstupní vrstvu. Těch už bylo o poznání méně, přesněji 401×6 . 401 je počet neuronů skryté vrstvy plus jedna konstanta. Hodnota 6 je počet výstupních neuronů ze sítě. První hodnoty pro každý řádek byly konstanty. Následovalo opět načítání hodnot po sloupcích až do konce souboru. Po kompletním zpracování tohoto souboru a načtení hodnot do proměnných jsem mohla přejít k rozhodování vícevrstvé neuronové sítě.

4.20.4 Rozhodování vícevrstvé neuronové sítě

Pro rozhodování ve vícevrstvé neuronové síti se musí rozpoznat, jaká váha bude použita. Ta se vybírá podle rozměru. Některým rozměrům odpovídá více vah, proto je pro rychlejší rozhodování důležité, zda se rozpoznávají mince napříč měnami nebo jen v rámci jedné měny. V případě jedné měny se použijí jen ty váhy, které odpovídají této měně. V druhém případě pro rozpoznávání mincí napříč měnami se použijí všechny váhy, které odpovídají danému rozměru.

Rozhodování pro vícevrstvé neuronové sítě je ze začátku stejné jako u Hopfieldovy sítě. To znamená, že prvně se vypočítají u . U vícevrstvé neuronové sítě se tato hodnota u počítá dvakrát, jednou pro výstup ze skryté vrstvy, podruhé pro výstup z neuronové sítě. Prvně se vypočítají hodnoty u pro výstup ze skryté vrstvy, což znamená vynásobení všech hodnot vah v řádku s odpovídající hodnotou vektoru k , který odpovídá obrazu pro rozpoznávání. První hodnota v daném řádku matice se vždy vynásobí s hodnotou jedna, jelikož je to hodnota konstanty vektoru k . Všechny tyto vynásobené hodnoty v řádku se sečtou a výsledkem je hodnota u pro daný řádek. Tento proces se opakuje pro každý řádek matice.

Po vypočítání hodnot u pro výstup ze skryté vrstvy se použije transformační funkce. U vícevrstvé neuronové sítě se využije *tansig* (hyperbolická tangenta přenosové funkce sigmoid). Tato transformační funkce má následující vzorec:

$$u_j = 2 / (1 + \exp(2 * -u_j)) - 1$$

kde j nám udává index ve vypočteném poli u .

Nyní následuje počítání hodnoty vektoru u pro výstup z neuronové sítě. Výpočet probíhá stejně, jako u výpočtu vektoru u pro výstup ze skryté vrstvy.

Posledním krokem pro rozhodování je transformační funkce pro výstup z neuronové sítě, která je proti předchozí transformační funkci zjednodušena. Nyní se jen kontroluje, zda je hodnota u na daném indexu větší či menší nule. Při hodnotě větší nebo rovna nule se do stejného indexu výstupu neuronové sítě $k+1$ vloží 1, při hodnotě nižší než 0 se vloží 0.

Nyní následuje jen porovnání hodnot vektoru výstupu neuronové sítě s požadovanými výstupními hodnotami přiřazenými k různým mincím. V případě shody se na výstupu programu zobrazí název mince, která se shoduje s rozpoznávaným obrazem. Tento název je důležitý pro aplikaci v Control webu k určení, u které minci se zvětší její zatříděný počet a kam se má daná mince umístit.

4.20.5 Výsledky vícevrstvé neuronové sítě

První testování neuronové sítě jsem aplikovala na fotky mincí získaných z České národní banky. Abych dosáhla potřebného počtu vzorů, které nejsou stejné, použila jsem generátor šumu obrazu a takto vytvořila 500 různých vzorů jedné mince. Už při učení této sítě v Matlabu jsem věděla, že tato metoda bude úspěšná. U těchto vzorů z české národní banky byla úspěšnost rozpoznávání vysoká. Z těchto výsledků nelze dělat závěry, jelikož tyto mince jsou ideálně nafocené a mince pořízené z modelu třídění mohou dávat jiné výsledky.

Testování na mincích nafocených modelem nebylo tak úspěšné, jako na fotkách mincí z České národní banky. U některých vzorků mince nebyla rozpoznána a mohla by být vytříděna jako mince jiné hodnoty. Tento špatný výsledek ale nebyl častý.

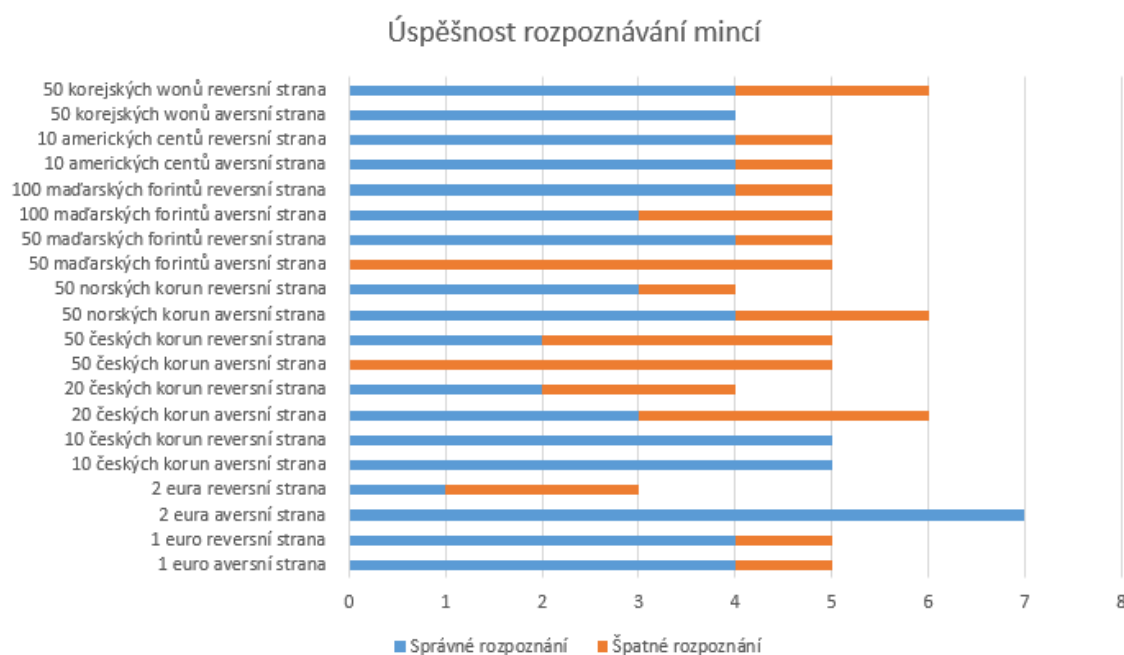
Pro testování jsem si vybrala 100 různých vzorů složených z 10 různých mincí různých měn. Správně bylo rozpoznáno 67 vzorů, to však neznamená, že rozpoznávání neuronové sítě má úspěšnost jen 67 %. Velké množství nerozpoznaných mincí je i proto, že nebyl nalezen správný rozměr mince, který byl očekáván pro výběr správné neuronové sítě. Z důvodu nenalezení správného rozměru nebylo rozpoznáno 18 mincí. Zbytek mincí (15) by bylo označeno za jinou hodnotu než má mince ve skutečnosti. Některé mince byly označeny jinou hodnotou stejné měny, například mince, které mají jednu stranu mince stejnou jako jiná mince. Nastaly i případy, kdy mince byla označena za minci úplně jiné hodnoty, jelikož neuronová síť se nemohla přiklonit ani k jedné minci, kterou měla rozpoznávat. Níže je zobrazena tabulka použitých vzorů pro rozpoznávání a úspěšnost jejich rozpoznávání. Ve druhém sloupci je množství správně rozpoznávaných mincí dané hodnoty, ve třetím sloupci nesprávně rozpoznávaných pomocí neuronové sítě a poslední sloupec udává množství mincí špatně rozpoznávaných z důvodu nesprávného rozměru. Pod tabulkou na obrázku 46 následuje grafické znázornění výsledků.

Název mince	SR	ŠNS	ŠUR
1 euro aversní strana	4	1	0
1 euro reversní strana	4	1	0
2 eura aversní strana	7	0	0
2 eura reversní strana	1	0	2
10 českých korun aversní strana	5	0	0
10 českých korun reversní strana	5	0	0
20 českých korun aversní strana	3	0	3
20 českých korun reversní strana	2	0	2
50 českých korun aversní strana	0	4	1
50 českých korun reversní strana	2	1	2
50 norských korun aversní strana	4	1	1
50 norských korun reversní strana	3	0	1
50 maďarských forintů aversní strana	0	5	0
50 maďarských forintů reversní strana	4	0	1
100 maďarských forintů aversní strana	3	2	0
100 maďarských forintů reversní strana	4	0	1
10 amerických centů aversní strana	4	0	1
10 amerických centů reversní strana	4	0	1
50 korejských wonů aversní strana	4	0	0
50 korejských wonů reversní strana	4	0	2
Celkem	67	15	18

Tabulka 5: Výsledky rozpoznávání nafocených mincí

SR - správné rozhodování, ŠNS - špatné rozhodování neuronové sítě, ŠUR - špatné určení rozměru

Dále jsem testovala rozpoznávání napříč měnami. Ale toto rozpoznávání nedává správné výsledky. Většinou při rozpoznávání mince, která má stejný rozměr pro více hodnot, vybere jako výsledek více hodnot mincí. Toto je dáno tím, že neuronová síť se učí mimo jiné i pomocí rozměru a v tomto případě jsou mince všechny stejné.



Obrázek 46: Grafické znázornění výsledků rozpoznávání nafocených mincí.

4.20.6 Závěr vícevrstvé neuronové sítě

Závěrem lze konstatovat, že tato metoda je vhodná pro rozpoznávání mincí v rámci jedné měny. Vykazovala skvělé výsledky pro rozpoznávání mincí z fotek České národní banky a 67% úspěšnost u nafocených mincí modelem. Z výsledků testování vyplývá, že horší výsledky jsou při rozpoznávání aversní strany mince, což se dalo očekávat z důvodu podobností této strany mezi hodnotami mincí. U této metody hodně záleží na počtu vzorů pro trénovací množinu. V případě malého množství se neuronová síť nemusí naučit rozpoznávat danou minci, což neznamená, že velké množství vzorů je vhodné. Zde zase hrozí přetrénování mince. Jelikož ve skryté vrstvě je 400 neuronů, zvolila jsem pro každou stranu mince přibližně 400 vzorů, což je polovina vzorů v jedné neuronové síti. Pro rozpoznávání mincí napříč měnami metoda vícevrstvé neuronové sítě vhodná není, jelikož vybírá více výsledků hodnot mincí za správné.

5 Závěr

V diplomové práci bylo potřeba nejdříve analyzovat metody pro rozpoznávání obrazu. Vybrala jsem dvě metody rozpoznávání obrazu pomocí deskriptorů (SIFT a SURF) a dvě metody s využitím umělých neuronových sítí (Hopfieldova síť a Vícevrstvá neuronová síť). Dále bylo nutno vyrobit nový třídící model s lepším osvětlením mince a s jeho dalším vylepšením. Jelikož se v modelu změnilo pár elektronických prvků a také se zvýšil počet tříděných mincí, bylo potřeba poupravit také aplikaci vytvořenou v Control webu.

Zlepšením osvětlení jde u mincí dobře rozpoznat reliéf při potřebné úpravě obrazu před rozpoznáváním. Dále byly do modelu přidány mikrospínače, které slouží k tomu, aby se třídící rameno nedostávala mimo povolený rozsah. Další velkou změnou oproti původnímu modelu bylo použití krokového motoru Microcon SX16-0402N. Ten umožnil lepší nastavení poloh pro třídění mincí do patřičných krabiček. Motor má ale slabý výkon a nemůže utáhnout rameno v potřebném rozsahu. I když je rameno vyrobeno z lehkého hliníku, v pravé polovině rozsahu se krokový motor přestává pohybovat ve směru doprava. Uvažovala jsem o využití plastového ramene, vyrobeného ze stejného plastu, jako je celý model, tedy plastu o tloušťce 5 mm. Ten by se však bez vyztužení ohýbal. Jeho vyztužení by ramenu zvýšilo váhu a opět by nebylo možné s ním pohybovat v celém rozsahu. Řešením by bylo zvýšení napájecího napětí krokového motoru. Použitý motor může být napájen v rozsahu 8 - 45 V. Pro napájení celého modelu je použit počítačový zdroj a tak je z něj motor napájen pouze 12 V.

Aplikace vytvořená v programu Control web byla upravena pro větší množství rozpoznávaných měn. Hlavní úprava spočívala ve výběru měn, kde jsem využila možnosti výběru Select boxem. Vedle Select boxu je ještě zobrazena vlajka státu, ke kterému se daná měna vztahuje. Z důvodu velkého množství mincí musel být upravený i archiver pro lepší čitelnost archivovaných souborů, kde je uložena historie tříděných mincí a jejich množství. Pro výběr měn, mezi kterými chce uživatel třídít jsem musela vytvořit tabulku, ve které se zobrazují uživateli jím vybrané měny. Po grafické stránce aplikace se změnilo pozadí aplikace a přemístilo se pár prvků pro lepší viditelnost aplikace.

V části rozpoznávání obrazu, hlavní náplní této diplomové práce, jsem implementovala a otestovala 4 metody. Metody SIFT a SURF, které rozpoznávají obraz pomocí deskriptorů, vykazovaly špatné výsledky, kdy ani jedna mince nebyla správně přiřazena ke svému vzoru. Důvodem bylo přiřazení většího počtu společných klíčových bodů s jinou mincí než se vzorem správné mince. Pak jsem přešla k metodám umělé neuronové sítě. Nejdříve jsem vyzkoušela Hopfieldovou síť, která vyhodnocovala správně jen v případě, že byla naučená pro velmi malý počet různých mincí. Výsledky byly špatné i v případě, že mince byla jakkoli natočená, jelikož tato metoda není robustní natočení obrazu. Poslední zvolenou metodou byla Vícevrstvá neuronová síť. Zvolila jsem dvouvrstvou neuronovou síť, kde vstupní vrstva měla 2500 vstupů, skrytá vrstva 400 neuronů a výstupní vrstva 6 neuronů. Učení probí-

halo pomocí programu Matlab s algoritmem backpropagation. Rozpoznávání jsem implementovala v programu Visual Studio a výsledná úspěšnost byla 67 %. Jelikož rozpoznávání někdy bylo trochu pomalé, protože v některých případech se provádělo rozhodování s několika váhami za sebou, jako vylepšení by bylo možné tuto část metody implementovat paralelně.

Ze všech 4 testovaných metod rozpoznávání obrazu se ukázala jako nejúspěšnější metoda Vícevrstvé neuronové sítě. Jako jediná měla úspěšnost rozpoznávání mincí nadpoloviční (67 %) a v případě, kdy bych počítala jen neúspěšnost v rozhodování neuronové sítě a ne neúspěšnost při špatném změření rozměru, výsledná úspěšnost by byla 88 %. Pro rychlejší rozpoznávání mincí se ukázalo nezbytné mít pro každou minci vlastní vícevrstvou neuronovou síť. Podle rozměru mince se určuje, která vícevrstvá neuronová síť bude použita.

6 Reference

- BÍLA, CSC., PROF. ING. JIŘÍ. *Umělá inteligence a neuronové sítě v aplikacích*. 1. Praha: Vydavatelství ČVUT, 1995. 115s..
- BÍLÝ, RADEK. *Control Web 2000*. 1. vyd. Praha: Computer Press, 1999. 382 s. ISBN 80-7226-258-0..
- ČUPOVÁ, JANA. *Třídění mincí pomocí strojového vidění*. [online]. Brno, 2013 [cit. 2015-04-15]. Dostupné z: Bakalářská práce. Mendelova univerzita v Brně. Vedoucí práce Ing. Richard Klein..
- DOBEŠ, M. *Zpracování obrazu a algoritmy v C#*. 1. vyd. Praha: BEN - technická literatura, 2008. 143 s. ISBN 978-80-7300-233-6..
- FAST Algorithm for Corner Detection. OpenCV 3.0.0-dev documentation*. [online]. 2014 [cit. 2016-04-11]. Dostupné z: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_fast/py_fast.html.
- FLANN - Fast Library for Approximate Nearest Neighbors*. [online]. 2012 [cit. 2016-03-28]. Dostupné z: <http://www.cs.ubc.ca/research/flann/>.
- Sift 0.3 documentation.. [online]. 2014 [cit. 2016-04-11]. Dostupné z: http://pythonhosted.org/sift_pyocl/sift.html
- Grayscale. WhatIs.com*. [online]. 2010 [cit. 2016-03-28]. Dostupné z: <http://whatis.techtarget.com/definition/grayscale>.
- Hardware: LifeCam Studio. Microsoft*. [online]. 2016 [cit. 2016-05-04]. Dostupné z: <https://www.microsoft.com/hardware/cs-cz/p/lifecam-studio>.
- HLAVÁČ, VÁCLAV. *Hledání hran*. [online]. In: . České vysoké učení technické v Praze, 2016 [cit. 2016-04-08]. Dostupné z: <http://cmp.felk.cvut.cz/hlavac/TeachPresCz/11DigZprObr/22EdgeDetectionCz.pdf>.
- HLAVÁČ, VÁCLAV. *Učení bez učitele*. [online]. In: . České vysoké učení technické v Praze, 2016 [cit. 2016-04-11]. Dostupné z: <http://cmp.felk.cvut.cz/hlavac/Public/TeachingLectures/UceniBezUcitele.pdf>.
- HLAVÁČ CSC., PROF. ING. VÁCLAV A DOC. ING. MILOŠ SEDLÁČEK, CSC.. *Zpracování signálů a obrazů*. Vyd. 1. Praha: Vydavatelství ČVUT, 2000. 220s. ISBN 80-01-02114-9..
- HRÚZ PH.D., ING. MAREK. *SIFT, SURF, MSER*. [online]. Plzeň, 2015 [cit. 2016-04-11]. Dostupné z: <http://www.kky.zcu.cz/uploads/courses/mpv/04/materialy04.pdf>.

- Introduction to SIFT. OpenCV: Open Source Computer Vision.* [online]. 2015 [cit. 2016-04-11]. Dostupné z: http://docs.opencv.org/3.1.0/da/df5/tutorial_py_sift_intro.html#gsc.tab=0.
- KAČENKA, PETR. *Neuronové sítě.* [online]. s. 12-16, 1998 [cit. 2016-04-11]. Dostupné z: <https://mks.mff.cuni.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf>.
- KOLOMAZNÍK, ING. JAN. *Aplikace moderních metod pro identifikaci obrazových dat [online].* Brno, 2014 [cit. 2015-11-26]. Dostupné z: https://is.mendelu.cz/auth/lide/clovek.pl?zalozka=7;id=21929;studium=59515;zp=36190;download_prace=1. Dizertační práce. Mendelova univerzita v Brně. Vedoucí práce Prof. RNDr. Ing. Jiří Štastný, CSc..
- LANDA, J. *Zpracování obrazu a jeho praktické využití.* [CD-ROM]. In PEFnet 2010: Evropská vědecká konference posluchačů doktorského studia. s. 47. ISBN 978-80-7375-450-1..
- MACEČEK, BC. ALEŠ. *Umělá neuronová síť RCE.* Brno, 2013. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Doc. Ing. VÁCLAV JIRSÍK, CSc..
- MALÝ, MAREK. *Vícevrstvé dopředné neuronové sítě: úvod do teorie a aplikací.* Vyd. 1. Ústí nad Labem: Univerzita J.E. Purkyně, Přírodovědecká fakulta, 2007. 72s. ISBN 978-80-7044-915-8..
- MARTIŠEK, D. *Matematické principy grafických systémů.* 1. vyd. Brno: Littera, 2002. 278 s. ISBN 80-85763-19-2..
- MathWorks.* [online]. 2016. [cit. 2016-05-04]. Dostupné z: <http://www.mathworks.com>.
- Modulární systém průmyslových vstupů/výstupů DataLab IO. Moravské přístroje.* [online]. Zlín - Malenovice, 2009 [cit. 2016-05-04]. Dostupné z: <http://www.mii.cz/art?id=134cat=77lang=405>.
- Programový systém Control Web. Moravské přístroje.* [online]. Zlín - Malenovice, 2016 [cit. 2016-05-04]. Dostupné z: <http://www.mii.cz/cat?id=146lang=405>.
- MUJA, MARIUS A DAVID LOWE. *FLANN - Fast Library for Approximate Nearest Neighbors: User Manual [online].* 2009 [cit. 2016-03-28]. Dostupné z: <http://www.cs.ubc.ca/~mariusm/uploads/FLANN/manual.pdf>.
- OLEJ, VLADIMÍR A PETR HÁJEK. *Úvod do umělé inteligence: moderní přístupy : distanční opora.* 1.vyd. Pardubice: Univerzita Pardubice, 2010. 98s. ISBN 978-80-7395-307-2..
- ONDRÁČEK, TOMÁŠ. *Adaptivní vícevrstvé neuronové sítě: Adaptive multilayer neural networks : zkrácená verze Ph.D. Thesis.* Brno: Vysoké učení technické,

- c2006. 32s.
ISBN 80-214-3126-1..
- OpenCV*. [online]. [cit. 2016-05-04]. Dostupné z: <http://opencv.org>.
- Servo Motor Overview. Oriental motor*. [online]. 2016 [cit. 2016-03-26]. Dostupné z: <http://www.orientalmotor.com/technology/articles/servo-motor-overview.html>.
- ŠNOREK, MIROSLAV. *Neuronové sítě a neuropočítače*. Vyd. 1. Praha: Vydavatelství ČVUT, 2002.
ISBN 80-01-02549-7..
- Stepper Motor Overview. Oriental motor*. [online]. 2016 [cit. 2016-04-26]. Dostupné z: <http://www.orientalmotor.com/technology/articles/stepper-motor-overview.html>.
- ŠVARC, CSC., DOC. ING IVAN. *Automatické řízení*. Brno: Akademické nakladatelství CERM Brno, 2011. 355 s.
ISBN: 978-80-214-4398- 3..
- Učení s učitelem. E-learningová učebnice: Matematická biologie*. [online]. Brno, 2016 [cit. 2016-04-11]. Dostupné z: <http://portal.matematickabiologie.cz/index.php?pg=analyza-a-hodnoceni-biologickyx-dat-umela-inteligence-neuronove-site-jednotlivy-neuron-adaptacni-dynamika-neuronu-uceni-s-ucitelem>.
- UHLÍŘ, IVAN. *Elektrické stroje a pohony*. Vyd. 2., přeprac. Praha: Nakladatelství ČVUT, 2007.
ISBN: 9788001037300..
- . Using the DRV8825 Stepper Motor Driver Carrier, High Current version. Robokits. [online]. 2016 [cit. 2016-03-25]. Dostupné z: <http://www.robokits.co.nz/using-DRV8825-stepper-motor-driver-carrier-high-current-version>
- VAVŘIŇÁK, ING. PETR A ING. ANNA ŘEHOVÁ. *Automatizace*. [online]. 2006 [cit. 2015-11-26]. Dostupné z: http://www.sse-najizdarne.cz/dokumenty/studijni_materialy/automatizace.pdf.
- . Visual Studio . [online]. 2016 [cit. 2016-05-04]. Dostupné z: <https://www.visualstudio.com>
- VOLNÁ, EVA. *Umělá inteligence: rozpoznávání vzorů v dynamických datech*. 1. vyd. Praha: BEN - technická literatura, 2014. ISBN 978-80-7300-497-2..

Seznam tabulek

1	Nastavení velikosti kroku krokového motoru	13
2	Seznam mincí	38
3	Technické parametry krokového motoru SX16-0402N	41
4	Rozdělení pinů konektoru	44
5	Výsledky rozpoznávání nafocených mincí	66
6	Seznam mincí.	81

Seznam obrázků

1	Krokový motor	10
2	Servomotor	11
3	Prahování - vlevo původní snímek, vpravo výsledek prahování	13
4	Prahování - hledání optimálního prahu	14
5	Prahování - výsledek lokálního prahování	14
6	Cannyho detektor hran	15
7	Metoda FAST - výběr kruhu	17
8	Metoda SIFT - znázornění oktáv	19
9	Metoda SIFT - hledání lokálního maxima	20
10	Metoda SIFT - výpočet keypoint descriptoru	21
11	Metoda SIFT	21
12	Metoda SURF - Výpočet součtu hodnot v obdélníkovém regionu \sum pomocí integrálního obrazu	22
13	Metoda SURF - Aproximace druhé derivace Gaussovy funkce podle osy x, osy y a xy	23
14	Metoda SURF - Konstrukce scale-space	24
15	Metoda SURF - změna měřítka	24
16	Metoda SURF - Haarova vlnka	25
17	Metoda SURF - Určení orientace, největší vektor leží ve výseči (žlutá barva), která určuje výslednou orientaci klíčového bodu	26
18	Metoda SURF - Vlevo je odezva na Haarovu vlnku v okolí klíčového bodu dle orientace a napravo výsledný deskriptor	26
19	Umělé neurony použité v Hopfieldové síti.	29
20	Vícevrstvá neuronová síť	33
21	Webová kamera Microsoft LifeCam Studio	37
22	Jednotka Datalab IO	37
23	Třídící model.	39
24	Držák krokového motoru.	41
25	Zapojení krokového motoru pomocí driveru DRV8825	42
26	Deska plošných spojů zapojení krokového motoru	42
27	Držák LED osvětlení	44
28	Konektor Canon s 15 piny.	44
29	Aplikace v Control webu s výběrem jedné měny.	46
30	Metoda FLANN s SURF - Výsledek porovnání nafocené 1 Libry se svým vzorem.	49
31	Metoda FLANN s SURF - Výsledek porovnání nafocené 1 Libry se vzorem 50 kurusů turecké liry.	49
32	Metoda FLANN s SURF - Výsledek porovnání nafocené 50 koruny se vzorem 20 švýcarských centů.	50
33	Metoda FLANN s SURF - Výsledek porovnání nafocené 50 koruny se vzorem 10 australských centů.	50

34	Metoda FLANN se SIFT - Výsledek porovnání nafocené 1 Libry se svým vzorem.	52
35	Metoda FLANN se SIFT - Výsledek porovnání nafocené 1 Libry se vzorem polských 2 groszů.	52
36	Metoda FLANN se SIFT - Výsledek porovnání nafocené 50 koruny se vzorem 10 agorot izraelského šekelu.	53
37	Metoda FLANN se SIFT - Výsledek porovnání nafocené 50 koruny se vzorem 50 koruny.	53
38	Úspěšné rozpoznání mince o hodnotě 1 Kč z reversní strany.	57
39	Úspěšné rozpoznání mince o hodnotě 10 Kč z reversní strany.	57
40	Úspěšné rozpoznání mince o hodnotě 20 Kč z reversní strany.	58
41	Neúspěšné rozpoznání mince o hodnotě 50 Kč z reversní strany.	58
42	Vstupní obraz v odstínech šedi a s nalezenými kruhy.	59
43	Vstupní obraz oříznutý a v binárních hodnotách.	60
44	Vstupní obraz s vymazaným pozadím.	60
45	Výsledky učení neuronové sítě.	63
46	Grafické znázornění výsledků rozpoznávání nafocených mincí.	67
47	Aplikace v Control webu s výběrem více měn.	81
48	Detail osvětlení.	82
49	Schéma zapojení.	82

7 Příloha

7.1 Seznam mincí

Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Americký dolar	USD	1 cent	19,05 mm
Americký dolar	USD	5 centů	21,21 mm
Americký dolar	USD	10 centů	17,91 mm
Americký dolar	USD	25 centů	24,26 mm
Americký dolar	USD	50 centů	30,61 mm
Americký dolar	USD	1 dolar	26,5 mm
Australský dolar	AUD	10 centů	23,6 mm
Australský dolar	AUD	20 centů	28,65 mm
Australský dolar	AUD	50 centů	36,65 mm
Australský dolar	AUD	1 dolar	25 mm
Australský dolar	AUD	2 dolary	20,5 mm
Brazilský real	BRL	1 centavo	17 mm
Brazilský real	BRL	5 centavos	22 mm
Brazilský real	BRL	10 centavos	20 mm
Brazilský real	BRL	25 centavos	25 mm
Brazilský real	BRL	50 centavos	23 mm
Brazilský real	BRL	1 real	27 mm
Britská libra	GBP	1 pence	20,3 mm
Britská libra	GBP	2 pence	25,9 mm
Britská libra	GBP	5 pence	18 mm
Britská libra	GBP	10 pence	24,5 mm
Britská libra	GBP	20 pence	21,4 mm
Britská libra	GBP	50 pence	27,3 mm
Britská libra	GBP	1 libra	22,5 mm
Britská libra	GBP	2 libry	28,4 mm
Bulharský lev	BGN	1 stotinka	16 mm
Bulharský lev	BGN	2 stotinky	18 mm
Bulharský lev	BGN	5 stotinky	20 mm
Bulharský lev	BGN	10 stotinky	18,5 mm
Bulharský lev	BGN	20 stotinky	20,5 mm
Bulharský lev	BGN	50 stotinky	22,5 mm
Bulharský lev	BGN	1 lev	24,5 mm
Česká koruna	CZK	1 Kč	20 mm
Česká koruna	CZK	2 Kč	21,5 mm
Česká koruna	CZK	5 Kč	23 mm

Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Česká koruna	CZK	10 Kč	24,5 mm
Česká koruna	CZK	20 Kč	26 mm
Česká koruna	CZK	50 Kč	27,5 mm
Čínský juan	CNY	1 fen	18 mm
Čínský juan	CNY	2 fen	21 mm
Čínský juan	CNY	5 fen	24 mm
Čínský juan	CNY	1 Jiao	19,03 mm
Čínský juan	CNY	5 Jiao	20,5 mm
Čínský juan	CNY	1 Yuan	25 mm
Dánský koruna	DKK	50 ore	21,5 mm
Dánský koruna	DKK	1 koruna	20,25 mm
Dánský koruna	DKK	2 koruny	24,5 mm
Dánský koruna	DKK	5 korun	28,5 mm
Dánský koruna	DKK	10 korun	23,35 mm
Dánský koruna	DKK	20 korun	27 mm
Euro	EUR	1 cent	16,25 mm
Euro	EUR	2 cent	18,75 mm
Euro	EUR	5 cent	21,25 mm
Euro	EUR	10 cent	19,75 mm
Euro	EUR	20 cent	22,25 mm
Euro	EUR	50 cent	24,25 mm
Euro	EUR	1 euro	23,25 mm
Euro	EUR	2 eura	25,75 mm
Filipínské peso	PHP	1 centavo	15,5 mm
Filipínské peso	PHP	5 centavos	15,5 mm
Filipínské peso	PHP	10 centavos	17 mm
Filipínské peso	PHP	25 centavos	20 mm
Filipínské peso	PHP	1 peso	24 mm
Filipínské peso	PHP	5 pesos	27 mm
Filipínské peso	PHP	10 pesos	26,5 mm
Honkongský dolar	HKD	10 centů	17,5 mm
Honkongský dolar	HKD	20 centů	19 mm
Honkongský dolar	HKD	50 centů	22,5 mm
Honkongský dolar	HKD	1 dolar	25,5 mm
Honkongský dolar	HKD	2 dolary	28 mm
Honkongský dolar	HKD	5 dolarů	27 mm
Honkongský dolar	HKD	10 dolarů	24 mm
Chorvatská kuna	HRK	1 lípa	17 mm
Chorvatská kuna	HRK	2 lípy	19 mm

Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Chorvatská kuna	HRK	5 líp	18 mm
Chorvatská kuna	HRK	10 líp	20 mm
Chorvatská kuna	HRK	20 líp	18,5 mm
Chorvatská kuna	HRK	50 líp	20,5 mm
Chorvatská kuna	HRK	1 kuna	22,5 mm
Chorvatská kuna	HRK	2 kuny	24,5 mm
Chorvatská kuna	HRK	5 kun	26,5 mm
Chorvatská kuna	HRK	25 kun	32 mm
Indická rupie	INR	50 paisů	22 mm
Indická rupie	INR	1 rupie	22 mm
Indická rupie	INR	2 rupie	25 mm
Indická rupie	INR	5 rupií	23 mm
Indická rupie	INR	10 rupií	27 mm
Indonéská rupie	IDR	50 rupií	20 mm
Indonéská rupie	IDR	100 rupií	23 mm
Indonéská rupie	IDR	200 rupií	25 mm
Indonéská rupie	IDR	500 rupií	27 mm
Indonéská rupie	IDR	1000 rupií	24,15 mm
Izraelský šekel	ILS	10 agorot	22 mm
Izraelský šekel	ILS	50 agorot	26 mm
Izraelský šekel	ILS	1 šekel	18 mm
Izraelský šekel	ILS	2 šekely	21,6 mm
Izraelský šekel	ILS	5 šekelů	24 mm
Izraelský šekel	ILS	10 šekelů	23 mm
Japonský jen	JPY	1 jen	20 mm
Japonský jen	JPY	5 jenů	22 mm
Japonský jen	JPY	10 jenů	23,5 mm
Japonský jen	JPY	50 jenů	21 mm
Japonský jen	JPY	100 jenů	22,6 mm
Japonský jen	JPY	500 jenů	26,5 mm
Jihoafriický rand	ZAR	10 centů	16 mm
Jihoafriický rand	ZAR	20 centů	19 mm
Jihoafriický rand	ZAR	50 centů	22 mm
Jihoafriický rand	ZAR	1 rand	20 mm
Jihoafriický rand	ZAR	2 randy	23 mm
Jihoafriický rand	ZAR	5 randů	26 mm
Jihokorejský won	KRW	10 won	18 mm
Jihokorejský won	KRW	50 won	21,6 mm
Jihokorejský won	KRW	100 won	24 mm

Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Jihokorejský won	KRW	500 won	26,5 mm
Kanadský dolar	CAD	1 cent	19,05 mm
Kanadský dolar	CAD	5 centů	20,2 mm
Kanadský dolar	CAD	10 centů	18,03 mm
Kanadský dolar	CAD	25 centů	23,88 mm
Kanadský dolar	CAD	50 centů	27,13 mm
Kanadský dolar	CAD	1 dolar	26,5 mm
Kanadský dolar	CAD	2 dolary	28 mm
Maďarský forint	HUF	5 forintů	21,2 mm
Maďarský forint	HUF	10 forintů	24,8 mm
Maďarský forint	HUF	20 forintů	26,3 mm
Maďarský forint	HUF	50 forintů	27,4 mm
Maďarský forint	HUF	100 forintů	23,8 mm
Maďarský forint	HUF	200 forintů	28,3 mm
Malajský ringgit	MYR	5 senů	17,78 mm
Malajský ringgit	MYR	10 senů	18,8 mm
Malajský ringgit	MYR	20 senů	20,6 mm
Malajský ringgit	MYR	50 senů	22,65 mm
Mexické peso	MXN	5 centů	15,5 mm
Mexické peso	MXN	10 centů	14 mm
Mexické peso	MXN	20 centů	15,3 mm
Mexické peso	MXN	50 centů	17 mm
Mexické peso	MXN	1 peso	21 mm
Mexické peso	MXN	2 pesos	23 mm
Mexické peso	MXN	5 pesos	25,5 mm
Mexické peso	MXN	10 pesos	28 mm
Norská koruna	NOK	1 koruna	21 mm
Norská koruna	NOK	5 korun	26 mm
Norská koruna	NOK	10 korun	24 mm
Norská koruna	NOK	20 korun	27,5 mm
Novozélandský dolar	NZD	10 centů	20,5 mm
Novozélandský dolar	NZD	20 centů	21,75 mm
Novozélandský dolar	NZD	50 centů	24,75 mm
Novozélandský dolar	NZD	1 dolar	23 mm
Novozélandský dolar	NZD	2 dolary	26,5 mm
Polský zlotý	PLN	1 groš	15,5 mm
Polský zlotý	PLN	2 groše	17,5 mm
Polský zlotý	PLN	5 grošů	19,5 mm
Polský zlotý	PLN	10 grošů	16,5 mm

Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Polský zlotý	PLN	20 grošů	18,5 mm
Polský zlotý	PLN	50 grošů	20,5 mm
Polský zlotý	PLN	1 zlotý	23 mm
Polský zlotý	PLN	2 zloté	21,5 mm
Polský zlotý	PLN	5 zlotých	24 mm
Rumunské nové leu	RON	1 ban	16,75 mm
Rumunské nové leu	RON	5 bani	18,25 mm
Rumunské nové leu	RON	10 bani	20,5 mm
Rumunské nové leu	RON	50 bani	23,75 mm
Ruský rubl	RUB	10 kopějek	17,5 mm
Ruský rubl	RUB	50 kopějek	19,5 mm
Ruský rubl	RUB	1 rubl	20,5 mm
Ruský rubl	RUB	2 rubl	23 mm
Ruský rubl	RUB	5 rublů	25 mm
Ruský rubl	RUB	10 rublů	22 mm
Singapurský dolar	SGD	5 centů	16,75 mm
Singapurský dolar	SGD	10 centů	18,5 mm
Singapurský dolar	SGD	20 centů	21 mm
Singapurský dolar	SGD	50 centů	23 mm
Singapurský dolar	SGD	1 dolar	24,65 mm
Švédská koruna	SEK	1 koruna	25 mm
Švédská koruna	SEK	5 korun	28,5 mm
Švédská koruna	SEK	10 korun	20,5 mm
Švýcarský frank	CHF	5 centů	17,15 mm
Švýcarský frank	CHF	10 centů	19,15 mm
Švýcarský frank	CHF	20 centů	21,05 mm
Švýcarský frank	CHF	1/2 franku	18,2 mm
Švýcarský frank	CHF	1 frank	23,2 mm
Švýcarský frank	CHF	2 franky	27,4 mm
Švýcarský frank	CHF	5 franků	31,45 mm
Thajský baht	THB	1 satang	15 mm
Thajský baht	THB	5 satangů	16,5 mm
Thajský baht	THB	10 satangů	17,5 mm
Thajský baht	THB	25 satangů	16 mm
Thajský baht	THB	50 satangů	18 mm
Thajský baht	THB	1 baht	20 mm
Thajský baht	THB	2 bahty	21,75 mm
Thajský baht	THB	5 bahtů	24 mm
Thajský baht	THB	10 bahtů	26 mm

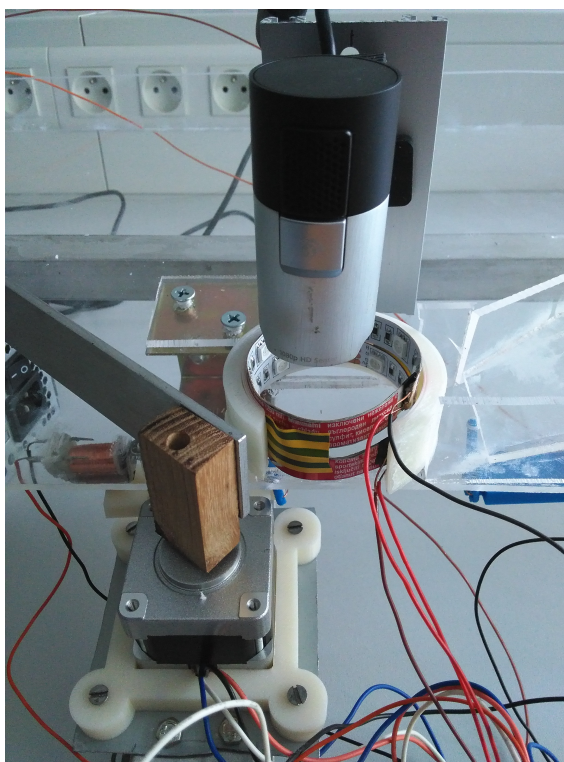
Seznam mincí různých měn			
Měna	Zkratka	Název mince	Rozměr
Turecká lira	TRY	1 kuruş	16,5 mm
Turecká lira	TRY	5 kuruş	17,5 mm
Turecká lira	TRY	10 kuruş	18,5 mm
Turecká lira	TRY	25 kuruş	20,5 mm
Turecká lira	TRY	50 kuruş	23,85 mm
Turecká lira	TRY	1 lira	26,15 mm

Tabulka 6: Seznam mincí.

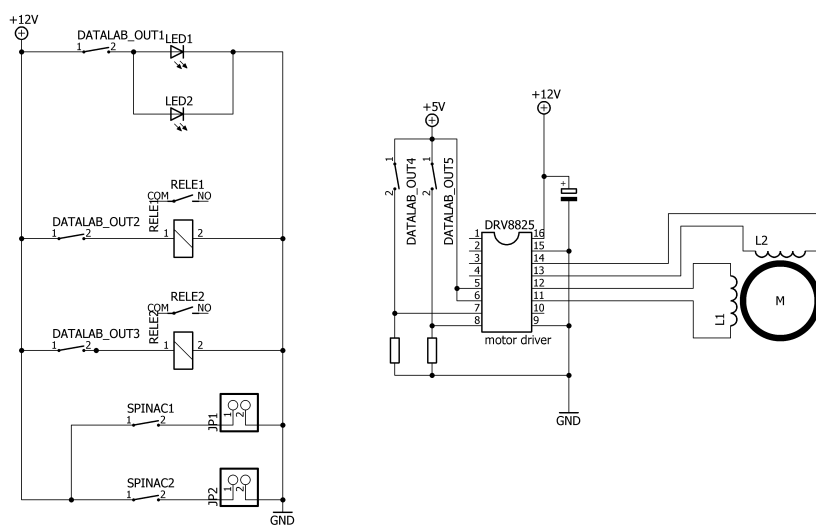
7.2 Přehled obrázků k praktické části



Obrázek 47: Aplikace v Control webu s výběrem více měn.



Obrázek 48: Detail osvětlení.



Obrázek 49: Schéma zapojení.