

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

DIPLOMOVÁ PRÁCE

Použití neuronových sítí v rozpoznávání objektů



Anotace

Práce seznamuje s využitím neuronových sítí v jednotlivých fázích rozpoznávání objektů v obraze. Podrobně se zaměřuje na fázi redukce dat pomocí analýzy hlavních komponent (PCA). Výstupem práce bude aplikace rozpoznávající objekty v obraze a sada experimentů s různými typy neuronových sítí použitých ve fázi redukce dat.

Děkuji vedoucímu diplomové práce RNDr. Eduardu Bartlovi, Ph.D. za cenné rady při tvorbě této diplomové práce. Dále děkuji své rodině za podporu během doby studia.

Obsah

1. Úvod	8
2. Základní pojmy a principy	9
2.1. Základy počítačové grafiky	9
2.2. Umělé neuronové sítě	10
2.2.1. Formální popis neuronu	11
2.2.2. Architektura neuronové sítě	12
2.2.3. Učení neuronové sítě	13
2.2.4. Hebbovo učení	14
2.2.5. Vybavování neuronové sítě	14
2.3. Analýza hlavních komponent PCA	15
2.3.1. Zaměření metody PCA	15
2.3.2. Princip metody analýzy hlavních komponent	15
2.3.3. Základní reprezentace dat	17
2.3.4. Redukce dimensionality	18
2.4. Využití neuronových sítí ve zpracování obrazu	19
2.4.1. Předzpracování	19
2.4.2. Redukce dat	19
2.4.3. Segmentace	20
2.4.4. Rozpoznání objektu	21
2.4.5. Porozumění obrazu	21
3. Neuronové sítě realizující PCA a rozpoznávání	23
3.1. Neuronová síť GHA	23
3.2. Neuronová síť APEX	24
3.3. Popis objektu	26
3.4. Aplikace rozpoznávání objektů v obraze	27
3.5. Přístup pomocí eigenfaces	28
3.6. Výpočet eigenfaces	30
3.7. Použití eigenfaces pro klasifikaci tváří	32
4. Uživatelská příručka	35
4.1. Popis aplikace	35
4.2. Požadavky na instalaci	35
4.3. Hlavní okno aplikace	35
4.4. Záložka Main	35
4.5. Záložka Settings	37
5. Testy	38
5.1. Popis vstupních dat	38
5.2. Příprava experimentů s neuronovými sítěmi GHA a APEX	38

5.3. Klasifikace v závislosti na změně počtu hlavních komponent sítě .	40
5.4. Klasifikace v závislosti na změně rychlosti učení sítě	41
5.5. Klasifikace v závislosti na změně počtu epoch	44
Závěr	46
Conclusions	47
Reference	48
6. Obsah přiloženého CD	49

Seznam obrázků

1.	Geometrická reprezentace modelu RGB.	9
2.	Popis neuronu.	11
3.	Sigmoida.	12
4.	Dopředná síť 4 – 2 a dopředná síť 4 – 3 – 2.	13
5.	Vzájemný vztah originálního vektoru x_i , rekonstruovaného vektoru \hat{x}_i a chybového vektoru e_i . Obrázek je převzat z http://latent.connectmv.com/wiki/Principal_Component_Analysis . 18	
6.	Dopředná síť s lineárními neurony ve výstupní vrstvě.	23
7.	Neuronová síť APEX.	25
8.	Schéma systému na rozpoznávání objektů.	28
9.	Transformace trénovací množiny do K vybraných eigenfaces.	29
10.	Reprezentace tváře pomocí lineární kombinace eigenfaces.	29
11.	Vypočítané eigenfaces z dané trénovací množiny.	30
12.	Trénovací množina, která obsahuje všechny obrazy stejné dimenze.	31
13.	PCA pro obrazy o rozměrech 1×2	32
14.	Schéma pro rozpoznávání tváří pomocí PCA.	33
15.	Aplikace euklidovské vzdálenosti pro rozpoznávání neznámého obrazu (zelený bod).	34
16.	Hlavní okno aplikace - záložka Main.	36
17.	Okno, které zobrazuje eigenvectors.	37
18.	Hlavní okno aplikace - záložka Settings.	37
19.	Příklad portrétových fotografií jediné konkrétní osoby.	38
20.	Vizualizace váhových vektorů pro síť GHA 0.0008 – 8 – 10.	43
21.	Vizualizace váhových vektorů pro síť GHA 0.8 – 8 – 10.	43

Seznam tabulek

1.	Výsledky rozpoznávání pro síť GHA s variabilním parametrem počet komponent.	41
2.	Výsledky rozpoznávání pro síť APEX s variabilním parametrem počet komponent.	41
3.	Výsledky rozpoznávání pro síť GHA s variabilním parametrem rychlost učení sítě.	43
4.	Výsledky rozpoznávání pro síť APEX s variabilním parametrem rychlost učení sítě.	43
5.	Výsledky rozpoznávání pro síť GHA s variabilním parametrem počet epoch.	45
6.	Výsledky rozpoznávání pro síť APEX s variabilním parametrem počet epoch.	45

1. Úvod

Tato práce se zabývá aplikováním neuronových sítí v číslicovém zpracování signálu s hlavním zaměřením na možnosti využití neuronových sítí při extrahování podstatné informace a rozpoznávání obrazových dat. Tyto prvky patří do oblasti počítačového vidění. Počítačové vidění je relativně nový obor. Oblast počítačového vidění (Computer Vision) patří dnes k nejprogresivněji se rozvíjejícím oblastem informatiky. Rozpoznávání objektů v obraze je velmi důležitým oborem souvisejícím se zpracováním dat a umělou inteligencí. V posledních letech dochází k prudkému vývoji v oblasti výpočetní techniky, což umožňuje další vývoj a vylepšení algoritmů pro automatické rozpoznávání objektů v obraze. Snahou je, co nejvíce přiblížit svět počítačů k myšlenkovým pochodům člověka, které probíhají v lidském mozku a ke schopnostem člověka učit se. Stejně jako člověk i počítač se musí naučit rozpoznávat objekty v daném prostředí na základě vizuální informace. Zdrojem těchto informací jsou pro počítač vstupní snímky.

V první fázi, která bývá označována jako *předzpracování obrazu*, se provede analýza vstupních obrazových dat. Zde se většinou aplikují metody skládající se z filtrace, geometrické normalizace, světelné normalizace apod. V další etapě *redukce dat* se uplatňují vedle tradičních metod (např. Analýza hlavních komponent (PCA)) stále více metody a algoritmy z oblasti umělé inteligence. Potom následuje fáze *segmentace*, ve které se obraz rozdělí na jednotlivé části (segmenty) se společnými znaky (např. barva). Proces označený jako *rozpoznávání objektů v obraze* určuje pozici, natočení a podobné vlastnosti objektu. Poslední fází, to je *porozumění obrazu* se využívá obou předchozích metod (segmentace a rozpoznávání objektů) s cílem získat co nejvíce informací o obsahu obrazu.

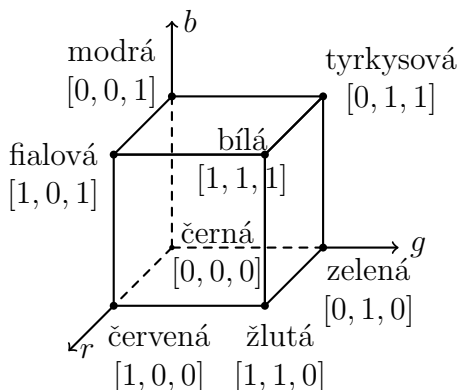
V procesu napodobit učení člověka se hojně využívají neuronové sítě, kterými lze řešit i problémy, které je člověk schopen jen obtížně popsat. Jejich výhodou je, že neuronová síť je schopna se učit z předložených dat i bez znalosti algoritmu pro řešení daného problému. Neuronové sítě tedy stačí znalost počtu příkladu a jejich řešení. Metody rozpoznávání objektů našly praktické uplatnění v celé řadě úloh.

Tato práce je rozčleněna do pěti kapitol. V úvodní části je popsána velmi úzká spojitost neuronových sítí s myšlenkovými pochody, které probíhají v mozku člověka při rozpoznávání obrazu. Ve druhé kapitole je čtenář obeznámen se základními pojmy a principy fungování umělých neuronových sítí a jejich následném využití v jednotlivých fázích rozpoznávání objektů v obraze. Třetí kapitola popisuje dva typy neuronových sítí pro extrakci příznaků z obrazu. Čtvrtá kapitola obsahuje uživatelskou příručku. V páté kapitole jsou uvedeny testy neuronových sítí *GHA* a *APEX*, které byly představeny v kapitole 3. Získané výsledky jsou shrnuty v závěrečné části této práce.

2. Základní pojmy a principy

2.1. Základy počítačové grafiky

Při psaní této podkapitoly jsem vycházel z knihy [15]. *Barevný model RGB* vychází z faktu, že lidské oko obsahuje tři základní druhy buněk citlivých na barvu. Tyto buňky jsou citlivé na elektromagnetické záření vlnové délky, které zhruba odpovídají červenému světlu (630 nm), zelenému (530 nm) a modrému (450 nm). Kombinací těchto tří barev *R* (červená), *G* (zelená) a *B* (modrá) lze získat téměř všechny barvy barevného spektra. Tento model využívá součtové (aditivní) skládání barev – čím více barev složíme (sečteme), tím světlejší je výsledek. Barevný model RGB lze vyjádřit pomocí jednotkové krychle, která je zobrazena na obrázku č. 1. Černá barva je umístěna v počátku na souřadnicích $[0, 0, 0]$. Na souřadnicích $[1, 1, 1]$ se nachází bílá barva. Jednotlivé osy, které jsou označeny *r*, *g* a *b* představují základní barvy. Zbývající barvy jsou doplňkové.



Obrázek 1. Geometrická reprezentace modelu RGB.

Barevná hloubka (*Color Depth*) někdy též bitová hloubka označuje počet bitů použitých pro uložení barvy každého kanálu v barevném modelu. Toto pojetí je také známé jako počet bitů na pixel. Jestliže barevný model používá např. tři základní barvy (např. model RGB), potom intenzita každé barvy je určena jedním číslem, které je vytvořeno určitým počtem bitů. Například 8bitová barevná hloubka používá 8 bitů na každou RGB barvu (na každý kanál) a jeden pixel tedy potřebuje $3 \cdot 8 = 24$ bitů = 3 byty. Větší barevná hloubka zvětšuje škálu různých barev (tzv. tonální rozsah) ale přirozeně také zvyšuje paměťovou náročnost obrázku.

Pro následné zpracování jednotlivých obrazů, které jsou postupně předkládány neuronové síti není potřeba zkoumat barevné složky obrazu odděleně. Z tohoto důvodu jsme jednotlivé barevné obrazy převedli na obrazy v odstínech šedi, kde odstíny šedi jsou vyjádřeny hodnotami jasu. Obrazy

v odstínech šedi se také nazývají *monochromatické*. Barevná hloubka monochromatického obrazu je 8 bitů. Z výše uvedeného plyne, že do každého pixelu monochromatického obrazu lze uložit hodnotu z rozsahu 0 až 255. Hodnota 0 uložená v pixelu reprezentuje černou barvu. Hodnota 255 reprezentuje bílou barvu.

Pro převod barevného obrazu v němž každý pixel obrazu je tvořen třemi složkami R (červená), G (zelená) a B (modrá) na obraz v odstínech šedi je možné použít následující vzorec

$$y_j = 0,299R_j + 0,587G_j + 0,114B_j,$$

kde y_j je výsledná hodnota jasu j -tého pixelu v šedotónovém obrazu. Hodnoty R_j , G_j a B_j jsou hodnoty jasů jednotlivých složek j -tého pixelu v barevném obraze.

2.2. Umělé neuronové sítě

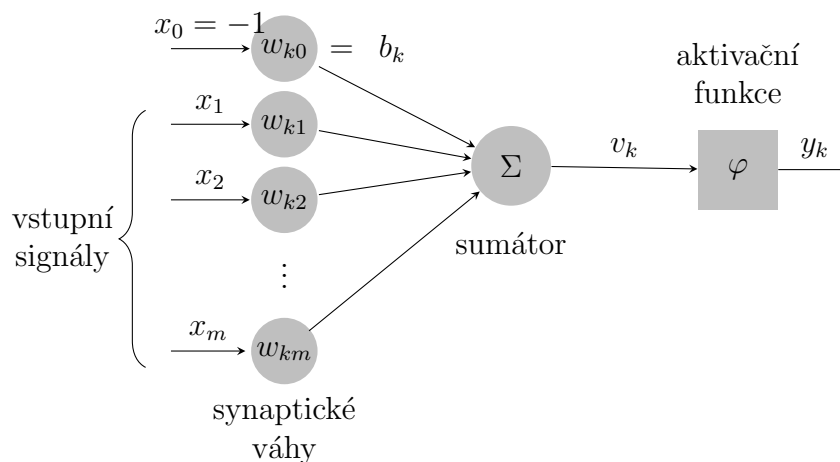
Při psaní této podkapitoly jsem především vycházel z knih [3], [4], [10], [12] a [14]. Umělé neuronové sítě jsou založeny na poznacích z neurofyziologie, která se zabývá funkcí nervových buněk a mozku živých organismů. Umělá neuronová síť je systém, který se skládá z výpočetních jednotek – neuronů. Tyto neurony jsou mezi sebou vzájemně propojeny. Z hlediska využití rozlišujeme v síti vstupní, skryté a výstupní neurony. Počet neuronů a jejich vzájemné propojení v síti určuje topologii neuronové sítě.

Neuronová síť se v čase vyvíjí, mění se stav neuronů, adaptují se váhy. Každá neuronová síť se dá rozdělit do tří částí: *organizační* (topologie sítě), *adaptivní* (učení sítě) a *aktivní* (vybavování sítě). Na základě toho pak mohou vzniknout zcela rozdílné modely neuronových sítí, které jsou vhodné pro řešení různých úloh. Velmi důležitou vlastností neuronových sítí je kromě schopnosti učit se, také schopnost *generalizace*. Schopnost učení neuronové sítě spočívá v nacházení různých závislostí v trénovacích datech. Tyto pak může definovat pomocí vah. Schopnost generalizace umožňuje neuronové síti správně reagovat i na neznámé vstupy, které nebyly předmětem tréninku.

Neuronové sítě našly v praxi celou řadu uplatnění. Lze je použít pro rozpoznávání (např. rozpoznávání znaků). Další možnou aplikací je řízení složitých zařízení v dynamicky se měnících podmínkách (např. autopilot automobilu). Lze je využít i v predikci a následném rozhodování (např. předpověď počasí, nebo vývoj cen akcií na burze). Může to být i analýza signálu (např. EKG, nebo EEG) a transformace signálů (např. převod anglicky psaného textu na mluvený signál). Neuronové sítě našly také svoje uplatnění v oblasti komprese dat.

2.2.1. Formální popis neuronu

Neuron tvoří základní výpočetní jednotku neuronové sítě. Na obrázku č. 2. je znázorněn model neuronu.



Obrázek 2. Popis neuronu.

Formální neuron má obecně m reálných vstupů x_1, \dots, x_m reprezentující *den-drity* [4]. Model neuronu obsahuje synapse (spojení). Všechny synapse jsou ohodnoceny reálnými synaptickými vahami w_{k1}, \dots, w_{km} neuronu k . Signál x_j na vstupu synapse j spojený s neuronem k je násobený synaptickou vahou w_{kj} . Hodnota synaptické váhy může nabývat kladné nebo záporné hodnoty. V případě kladné hodnoty se toto spojení označuje jako *excitační spojení*. V případě záporné hodnoty se jedná o *inhibiční spojení*.

Další nedílnou součástí modelu neuronu je sumátor, který reprezentuje *tělo buňky* [4]. Ten bývá v odborné literatuře označován jako *vnitřní potenciál* v_k neuronu k . Jedná se o váženou sumu vstupních hodnot s příslušnými synaptickými vahami.

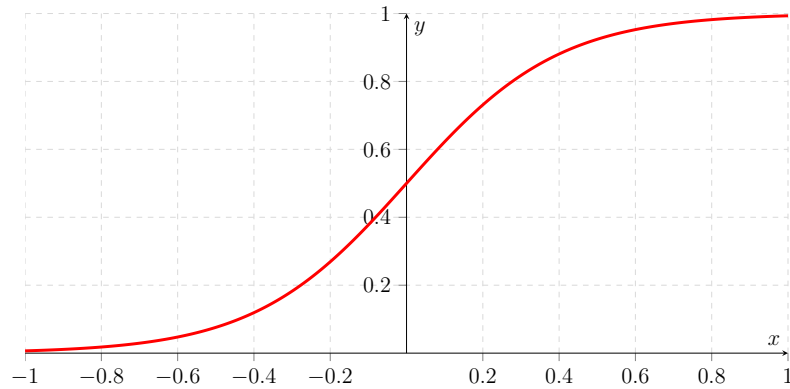
Model neuronu obsahuje také práh b_k . Ten je přidán jako speciální vstup $x_0 = 1$ k vstupním signálům x_1, \dots, x_m . S ním se potom pracuje jako s jakoukoliv jinou synaptickou vahou. Ten zaručuje zvýšení nebo snížení vstupu do aktivační funkce v závislosti na tom, zda je jeho hodnota kladná nebo záporná.

Potom následuje aktivační funkce φ . Aktivační funkce se používá pro převod *vnitřního potenciálu* v_k neuronu do normalizovaného rozsahu výstupu y_k neuronu. Tím je modelován elektrický impuls *axonu* [4]. Typický normalizovaný rozsah výstupu y_k neuronu je z intervalu $[0, 1]$ nebo $[-1, 1]$. Základním typem aktivační funkce může být prahová funkce, po částech lineární funkce nebo sigmoida.

Sigmoida je nejčastější aktivační funkcí používanou při konstrukci umělých neuronových sítí. Tato funkce je definována pomocí rovnice

$$f(x) = \frac{1}{1 + e^{-ax}}, \quad (2.2.1)$$

kde a je parametr ovlivňující strmou funkci. Hodnoty funkce se blíží k 0 v $-\infty$. Hodnota funkce se blíží k 1 v $+\infty$. Pro vstupní hodnotou funkce 0 obdržíme funkční hodnotu 0.5. Tato funkce vrací pouze kladné hodnoty. Graf této funkce (2.2.1) je zobrazen na obrázku č. 3.



Obrázek 3. Sigmoida.

V naší aplikaci jsme využili lineární aktivační funkci. Tato funkce je definována předpisem

$$f(x) = \frac{x}{3456},$$

kde číslo 3456 reprezentuje dimenzi obrázku.

Matematicky se dá k -tý neuron popsat následovně pomocí rovnic

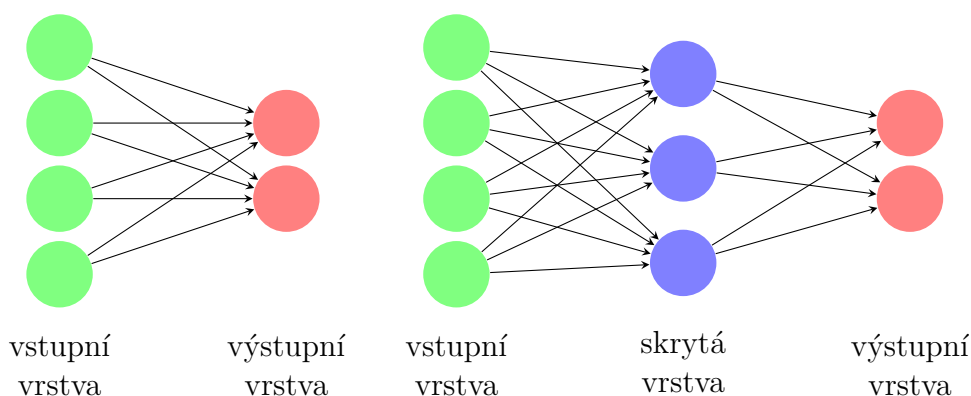
$$v_k = \sum_{j=0}^m w_{kj} x_j,$$

$$y_k = \varphi(v_k).$$

2.2.2. Architektura neuronové sítě

Organizační dynamika neuronové sítě nám určuje architekturu neuronové sítě. Architektura neuronové sítě je vždy definována a je neměnná v čase. Existují však neuronové sítě, které umožňují během adaptivní fáze měnit svoji topologii. V případě potřeby je původní topologie doplněna o další neurony a spojení. Všeobecně rozeznáváme 2 typy topologií – *acyklická* (dopředná) a *cyklická* (rekurentní).

Acyklická topologie – Síť s takovou topologií neobsahuje cyklus a navíc všechny spojení vedou jedním směrem. Do tohoto druhu topologie spadají *vícevrstvé dopředné sítě*. Architektura těchto sítí je organizována ve formě vrstev. Každá vrstva je tvořena daným počtem neuronů. V nejjednodušším případě je *vstupní vrstva* neuronů spojenou s *výstupní vrstvou*. Na obrázku č. 4. je znázorněna jednovrstvá dopředná síť. V odborné literatuře se pro tento typ architektury používá označení $p - q$, kde p udává počet neuronů ve *vstupní vrstvě* a q udává počet neuronů ve *výstupní vrstvě*. Další variantou vícevrstvé dopředné sítě je architektura, která obsahuje kromě *vstupní vrstvy* a *výstupní vrstvy* neuronů jednu nebo více *skrytých vrstev*.



Obrázek 4. Dopředná síť 4 – 2 a dopředná síť 4 – 3 – 2.

Cyklická topologie – Tento druh topologie neuronové sítě se vyznačuje tím, že zde existuje skupina neuronů, která je spojena v kruhu (tedy tvoří cyklus). Tato topologie je využívána především v *rekurentních neuronových sítích*. Od acyklické topologie se liší tím, že zde existuje alespoň jedna zpětná vazba.

2.2.3. Učení neuronové sítě

Učení neuronové sítě bývá v odborné literatuře označováno jako *adaptivní dynamika*. Před zahájením procesu učení se provede inicializace synaptických vah sítě na malé náhodné hodnoty z daného intervalu. Požadovaná funkce sítě je obvykle zadána tzv. *trénovací množinou*. Potom dochází k vlastnímu procesu učení sítě. V průběhu učení dochází ke změně hodnot těchto vah. Cílem učení je nastavit jednotlivé váhy spojení w_{ij} tak, aby síť vytvářela správnou odezvu na předložené vstupy z trénovací množiny.

Učení neuronové sítě se dělí na *učení s učitelem* (supervised learning) nebo *učení bez učitele* (unsupervised learning). Při *učení s učitelem* dochází v neuronové síti (to je v procesu učení) k vzájemnému srovnávání aktuálního výstupu s požadovaným výstupem. Synaptické váhy se potom nastaví tak, aby se snížil rozdíl mezi skutečným a požadovaným výstupem. Tento rozdíl je označován jako

chyba. Při *učení bez učitele* trénovací množina obsahuje pouze vstupy sítě. Synaptické váhy se potom nastavují tak, aby síť poskytovala stejnou odezvu při stejných, nebo podobných vstupních vektorech.

2.2.4. Hebbovo učení

Nejstarší a nejjednodušší pravidlo pro učení neuronové sítě je *Hebbovo pravidlo*. Jeho autorem je kanadský psycholog a neuropsycholog Donald Hebb. V roce 1949 napsal Donald Hebb knihu *Organization of Behavior*, ve které navrhl učící pravidlo samostatného neuronu. Toto pravidlo je dnes známe pod názvem *Hebbovo učení*.

Hebbovo učení je založeno na dvou pravidlech:

1. Pokud dva neurony v neuronové síti zahoří současně, vazbu mezi těmito neurony posílíme.
2. Pokud zahoří pouze jeden z neuronů, vazbu mezi těmito neurony oslabíme.

Matematický popis *Hebbova pravidla učení* je vyjádřen pomocí následující rovnice

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n),$$

kde $\Delta w_{kj}(n)$ vyjadřuje změnu synaptické váhy spojení z neuronu k do neuronu j v čase n . Dále $x_j(n)$ je vstupní vektor, $y_k(n)$ je výstupní vektor a η je parametr rychlosti učení. Toto adaptační pravidlo se potom použije pro všechny váhy.

2.2.5. Vybavování neuronové sítě

Pro proces vybavování neuronové sítě se v odborné literatuře používá termín *aktivní dynamika*. Tento proces vybavování neuronové sítě nastává až je neuronová síť naučená. Zde se vezme naučená neuronová síť a *testovací množina*. Jednotlivé vzorky jsou postupně přiváděny na vstup naučené neuronové sítě a zjišťuje se jejich odezva.

Množina vstupních dat je tvořena jednotlivými vzorky. Tyto vzorky jsou postupně předkládány neuronové síti. Tato množina dat se zpravidla rozděluje na dvě části – *testovací množinu* a *trénovací množinu*. Nejdůležitější je trénovací množina. *Trénovací množina* musí obsahovat vzorky, které nejlépe popisují vstupní množinu. Význam trénovací množiny určuje rychlost a kvalitu učení neuronové sítě. Jak moc je neuronová síť kvalitně naučena se provádí pomocí testovací množiny. *Testovací množina* by neměla obsahovat totožné vzorky, které jsou obsaženy v trénovací množině. Kdyby tomu tak bylo, tak by jsme nemohli ověřit kvalitu naučení neuronové sítě.

2.3. Analýza hlavních komponent PCA

Při zpracování této podkapitoly, která se týká metody analýzy hlavních komponent (PCA) jsem čerpal z publikací [1], [4] a [8].

2.3.1. Zaměření metody PCA

Tuto metodu představil Pearson v roce 1901. V celé řadě úloh se setkáváme s tím, že původní počet naměřených znaků je značný. Tím se nabízí otázka, jak takové množství znaků přehledně interpretovat. Pro jednodušší interpretaci nám metoda PCA provede vyjádření (lineární transformaci) původních znaků na menší počet nových znaků s co nejmenší ztrátou informace. Výhody takto nově získaných znaků mají zásadní vliv na zjednodušení konečné analýzy a vyhodnocení získaných výsledků měření. Takto získané nové znaky se označují jako hlavní komponenty. Dále jsou vhodnější pro další použití, protože je jich výrazně méně než původních znaků. Tyto nové znaky vystihují téměř celou proměnlivost (rozptyl) původních znaků a jsou vzájemně nekorelovatelné. Nové znaky, které jsme získali z původních naměřených znaků jsou jejich lineární kombinací. Nejvíce informací o rozptylu původních dat popisuje první hlavní komponenta. Druhá hlavní komponenta popisuje informace o rozptylu, které nejsou obsaženy v první hlavní komponentě atd. Poslední komponenta představuje nejmenší množství informací o rozptylu. Analýza hlavních komponent (PCA) našla svoje uplatnění nejvíce v oblasti analýzy vícerozměrných dat – redukce dimenze. Mimo tuto oblast je možné také využít metodu PCA v oblastech průzkumové analýzy dat [8] a testu vícerozměrné normality [8].

2.3.2. Princip metody analýzy hlavních komponent

Základní pojmy z oblasti matematické statistiky použité v této práci jsou vysvětleny v knize [1]. Cílem je nalézt vhodnou lineární transformaci, která bude optimální z pohledu minimalizace střední kvadratické chyby mezi zrekonstruovaným a originálním vektorem dat. Nechť $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ je náhodný vektor dimenze m . Tento vektor \mathbf{x} reprezentuje vstupní data. Předpokládejme, že náhodný vektor \mathbf{x} má nulovou střední hodnotu, tedy platí

$$E[\mathbf{x}] = [E[x_1], E[x_2], \dots, E[x_m]]^T = 0.$$

Pokud $E[\mathbf{x}] \neq 0$, nabízí se odečíst střední hodnotu z dat před analýzou. Nechť $\mathbf{u} = [u_1, u_2, \dots, u_m]^T$ je jednotkový vektor dimenze m , na který budeme provádět projekci vektoru \mathbf{x} . Tato projekce vektoru \mathbf{x} na jednotkový vektor \mathbf{u} je definována následovně

$$a = \mathbf{u}^T \mathbf{x} = \mathbf{x}^T \mathbf{u}. \quad (2.3.1)$$

Jedná se tedy o skalární součin vektorů \mathbf{x} a \mathbf{u} . Pro vektor \mathbf{u} platí, že $\|\mathbf{u}\| = \sqrt{\mathbf{u}^T \mathbf{u}} = 1$. Statistické vlastnosti projekce a jsou střední hodnota a rozptyl. Pro

její střední hodnotu platí, že

$$E[a] = E[\mathbf{u}^T \mathbf{x}] = \mathbf{u}^T E[\mathbf{x}] = 0.$$

Pro její rozptyl (varianci) platí, že

$$\sigma^2 = E[a^2] = E[(\mathbf{u}^T \mathbf{x})(\mathbf{x}^T \mathbf{u})] = \mathbf{u}^T E[\mathbf{x}\mathbf{x}^T] \mathbf{u} = \mathbf{u}^T \mathbf{R}_x \mathbf{u}.$$

Čtvercová matice \mathbf{R}_x typu $m \times m$ je korelační matice náhodného vektoru \mathbf{x} . Tato matice je definována jako

$$\mathbf{R}_x = E[\mathbf{x}\mathbf{x}^T] = \mathbf{R}_x^T.$$

Navíc korelační matice \mathbf{R}_x je symetrická. To znamená, že platí $\mathbf{R}_x^T = \mathbf{R}_x$.

Podstatná charakteristika daných dat \mathbf{x} může být množina m vlastních hodnot $\lambda_1, \lambda_2, \dots, \lambda_m$ a odpovídající vlastní vektory u_1, u_2, \dots, u_m vyřešením problému vlastních hodnot pomocí následující rovnice

$$\mathbf{R}_x \mathbf{u}_j = \lambda_j \mathbf{u}_j, \quad j = 1, 2, \dots, m. \quad (2.3.2)$$

Odpovídající vlastní hodnoty jsou uspořádány v sestupném pořadí $\lambda_1 > \lambda_2 > \dots > \lambda_m$ tak, že $\lambda_1 = \lambda_{MAX}$. Odpovídající ortonormální vlastní vektory $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m$ budou poskládány do matice typu $m \times m$

$$\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m].$$

Rovnice problému vlastních hodnot z (2.3.2) můžeme zapsat v maticovém tvaru jako

$$\mathbf{R}_x \mathbf{U} = \mathbf{U} \Lambda, \quad (2.3.3)$$

kde Λ je diagonální matice obsahující na hlavní diagonále vlastní čísla matice R_x . Diagonální matice Λ je ve tvaru

$$\Lambda = \text{diag}[\lambda_1 > \lambda_2 > \dots > \lambda_m].$$

Matice \mathbf{U} je ortogonální matice se sloupcovými vektory u_1, u_2, \dots, u_m , které jsou vlastními vektory matice \mathbf{R}_x a jsou ortonormální, jelikož splňují následující podmínku

$$\mathbf{u}_i^T \mathbf{u}_j = \begin{cases} 1 & j = i \\ 0 & j \neq i \end{cases}.$$

Protože pro ortogonální matici \mathbf{U} máme

$$\mathbf{U}^T \mathbf{U} = \mathbf{I},$$

kde \mathbf{I} je jednotková matice, potom inverzní matice k matici \mathbf{U} se rovná transponované matici \mathbf{U}

$$\mathbf{U}^T = \mathbf{U}^{-1}.$$

Z rovnice (2.3.3) můžeme vytvořit tzv. *ortogonální transformaci podobnosti*

$$\mathbf{U}^{-1}\mathbf{R}_x\mathbf{U} = \mathbf{U}^T\mathbf{R}_x\mathbf{U} = \Lambda. \quad (2.3.4)$$

Ortogonální transformace podobnosti transformuje korelační matici \mathbf{R}_x na diagonální matici vlastních hodnot.

2.3.3. Základní reprezentace dat

Protože existuje m různých jednotkových vektorů, můžeme definovat m různých projekcí na tyto vektory. Z rovnice (2.3.1)

$$a_j = u_j^T x = x^T u_j, \quad j = 1, 2, \dots, m$$

kde a_j jsou projekce vektoru x na hlavní směry, které jsou reprezentovány jednotkovými vektory u_j . Potom a_j se nazývají *hlavní komponenty*. Tyto projekce a_j pro $j = 1, 2, \dots, m$ uspořádáme do vektoru

$$\mathbf{a} = [a_1, a_2, \dots, a_m]^T. \quad (2.3.5)$$

Dále platí

$$\begin{aligned} \mathbf{a} &= [a_1, a_2, \dots, a_m]^T \\ &= [x^T u_1, x^T u_2, \dots, x^T u_m]^T \\ &= \mathbf{U}^T x. \end{aligned}$$

Z uvedeného vyplývá, že a_1 je tzv. *první hlavní komponenta*. Tedy projekce na vlastní vektor korelační matice, který odpovídá největšímu vlastnímu číslu korelační matice. Dále a_2 se nazývá *druhá hlavní komponenta*. Ta vznikne jako projekce na vlastní vektor, který odpovídá druhému největšímu vlastnímu číslu korelační matice. Dále a_3 se označuje jako *třetí hlavní komponenta*. Ta vznikne jako projekce na vlastní vektor, který odpovídá třetímu největšímu vlastnímu číslu korelační matice. Dále pokračujeme analogicky až do m .

Rekonstrukce originálního vektoru x z projekcí a_j pro $j = 1, 2, \dots, m$ se dá zapsat ve tvaru

$$\begin{aligned} x &= \mathbf{U}\mathbf{a} \\ &= \sum_{j=1}^m a_j u_j. \end{aligned} \quad (2.3.6)$$

Jednotkové vektory u tvoří bázi prostoru dat. Rovnice (2.3.6) představuje transformaci souřadnic, podle které bod x v prostoru dat je transformován do odpovídajícího bodu a v prostoru příznaků.

2.3.4. Redukce dimensionality

Metoda analýzy hlavních komponent může být efektivně použita pro extrakci příznaků a redukci dimenzionality. PCA je nejlepší technika pro lineární extrakci příznaků z původního datasetu ve smyslu minimalizace rekonstrukční chyby. Zejména můžeme snížit počet příznaků potřebných pro efektivní reprezentaci dat tím, že zanedbáme ty lineární kombinace v rovnici (2.3.6), které mají malé rozptyly a zachováme pouze ty, které mají větší rozptyly. Označme $\lambda_1, \lambda_2, \dots, \lambda_l$ jako l největších vlastních hodnot korelační matice R . Můžeme psát, že

$$\hat{x} = \sum_{j=1}^l a_j u_j, \quad l \leq m.$$

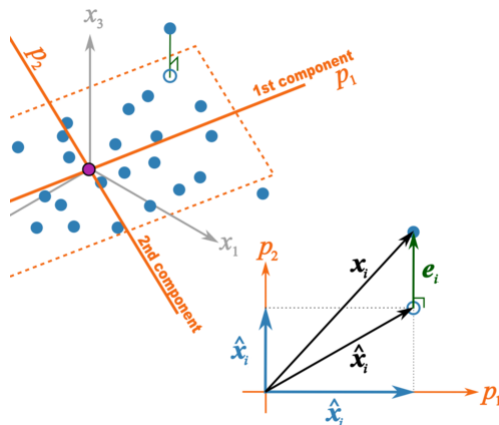
Jedná se vlastně o aproximaci vektoru x vektorem \hat{x} . Vektor a (2.3.5) byl tedy ořezán o prvky, které pořadím odpovídají nejmenším vlastním číslům. *Aproximační chybový vektor* e je definován

$$e = x - \hat{x}.$$

Z výše uvedeného je zřejmé, že vektor e je roven rozdílu mezi původním datovým vektorem x a aproximovaným datovým vektorem \hat{x} . Platí také

$$e = \sum_{j=l+1}^m a_j u_j.$$

Vektor aproximačních chyb e je ortogonální k rekonstruovanému vektoru \hat{x} . Toto je znázorněno na obrázku č. 5.



Obrázek 5. Vzájemný vztah originálního vektoru x_i , rekonstruovaného vektoru \hat{x}_i a chybového vektoru e_i . Obrázek je převzat z http://latent.connectmv.com/wiki/Principal_Component_Analysis.

2.4. Využití neuronových sítí ve zpracování obrazu

V této podkapitole, která se týká využití neuronových sítí v jednotlivých fázích rozpoznávání objektů v obraze vycházím z článku [2].

2.4.1. Předzpracování

Prvním krokem ve zpracování obrazu pomocí neuronových sítí je *předzpracování*. Operace pro předzpracování obecně spadají do jedné ze tří kategorií – *rekonstrukce obrazu* (Image reconstruction), *obnovení obrazu* (Image restoration) a *zvýraznění obrazu* (Image enhancement).

Během procesu *rekonstrukce obrazu* (Image reconstruction) jde o sestavení obrazu z počtu naměřených dat.

V části *obnovení obrazu* (Image restoration) se provádí odstranění šumu a vad, které vznikly během pořízení obrazu. Nejjednodušším způsobem obnovení obrazu je odstranění šumu pomocí filtrace. K filtraci se využívají různé typy neuronových sítí, kterými jsou popsány různé druhy filtrů. Potlačení šumu je možné provést také pomocí konvoluce. Jedná se o dopřednou neuronovou síť a dopřednou síť realizující regresi. Pro obnovu barevných obrazů a jejich převod na šedotónové obrazy se používá neuronová síť, která se označuje jako *CNN* (*Cellular Neural Network*). Další zajímavou architekturou neuronové sítě je *GANF* (*Generalized Adaptive Neural Filter*), která je také vhodná pro potlačení šumu.

Během procesu *zvýraznění obrazu* (Image enhancement) dochází k zvýraznění určitých vlastností, které mají za následek lepší segmentaci a rozpoznávání obrazu. K vylepšení obrazu se využívá detekce hran. V této části byly použity dopředné neuronové sítě, které realizovaly detekci hran. K tomu slouží například Hopfieldova síť, nebo regresní dopředná neuronová síť.

2.4.2. Redukce dat

Dvě z nejdůležitějších aplikací redukce dat jsou *kompresie* a *extrakce příznaků*. Algoritmy pro kompresi používáme pro uložení a přenos obrázků. Tento proces se skládá ze dvou kroků – *kompresie* a *dekompresie*. Pro oba tyto kroky je možné využít neuronové sítě. Pro kompresi obrazu je možné použít *dopředné neuronové sítě*, *SOM* (Samoorganizující se mapy) a *RBF* (Radial Basis Function).

Extrakce příznaků se používá pro následnou segmentaci nebo pro rozpoznávání objektů v obraze. Na extrakci příznaků se můžeme dívat jako na určitý druh redukce dat. Cílem tohoto procesu je nalézt podmnožinu příznaků na základě obrazových dat. Je zřejmé, že obrazová data mají značně vysokou dimenzi. Pro jejich další zpracování se nám obrazová data s takto velmi vysokou dimenzí nehodí. Při zpracování těchto dat by totiž došlo k značnému nárůstu doby výpočtu. Abychom se této nepříjemné vlastnosti vyvarovali, provedeme redukci dat pomocí *extrakce příznaků*. K dispozici máme celou řadu neuronových sítí, které provádí redukci dat. Uvedeme zde pouze např. *SOM* a *Hopfieldova síť*.

Touto fází se budeme podrobně zabývat v kapitole 3. Pro redukcí dat v této fázi zpracování obrazu jsme použili neuronové sítě *GHA* (Generalized Hebbian Algorithm) a *APEX* (Adaptive Principal Component Extraction), které řeší metodu *PCA* (Principal Component Analysis).

2.4.3. Segmentace

Hlavním úkolem segmentace je rozdělit obraz do takových částí (segmentů), které mají úzkou souvislost s oblastmi reálného světa, zachyceného na obraze, nebo mají společné vlastnosti. Mezi takové vlastnosti patří např. barva, tvar nebo textura. Výsledkem segmentace je soubor vzájemně se nepřekrývajících oblastí. Tyto výsledky jsou pak používány například pro rozpoznávání, nebo identifikaci objektů. Segmentace obrazu pomocí neuronových sítí je založena na dvou přístupech – *segmentace na základě pixelů* (based on pixel) a *segmentace na základě lokálních vlastností obrazu pomocí příznaků* (based on features).

První přístup je *segmentace na základě pixelů* (based on pixel). Tento přístup využívá znalost vlastnosti samostatného pixelu. Tento přístup je založen na principu, kdy je obraz jako celek přiveden na vstup neuronové sítě. Neuronová síť začne provádět segmentaci. Po jejím dokončení získáme jednotlivé segmenty, které byly součástí vstupního obrazu. Velkou nevýhodou tohoto přístupu je náchylnost na změnu měřítka a změnu rotace. Tato nevýhoda může zhoršit výsledek segmentace. Segmentaci založenou na základě pixelů lze realizovat různými typy neuronových sítí, jako jsou *dopředné neuronové sítě*, *SOM*, *CNN*, *Hopfieldovy sítě*, *pravděpodobnostní neuronové sítě* a další.

Výše uvedené sítě mohou být organizovány v určité hierarchii. Tento způsob organizace je založen na úrovních a postupuje se zde vždy směrem zdola nahoru (bottom-up). V každé úrovni se nachází neuronová síť, která se specializuje na hledání určitých vlastností ze svého vstupu. Po dokončení dané úrovně se provede předání částečného výsledku segmentace obrazu do následující úrovně. Zde se nachází další neuronová síť, která se zabývá hledáním zase dalších vlastností ze svého vstupu. Tak se pokračuje ve směru zdola - nahoru. Na nejvyšší úrovni se nachází neuronová síť, která se postará o spojení všech dílčích podvýsledků segmentace do jednoho celku. Tím je tedy segmentace obrazu hotová. Uvedené neuronové sítě jsou natrénovány na segmentaci na základě textury, nebo na základě kombinace textury a tvaru.

Druhý přístup je *segmentace na základě lokálních vlastností obrazu pomocí příznaků* (based on features). Zkoumá vlastnosti v okolí pixelu. Tento druh segmentace obrazu není náchylný na měřítka a rotaci. K tomu se dají využít následující typy neuronových sítí – *dopředná neuronová síť*, *rekurzivní neuronové sítě*, *SOM*, *Hopfieldovy sítě*, varianty *RBF sítí* a síť pro analýzu komponent.

Můžeme zde využít také hierarchickou architekturu neuronových sítí. Jsou založeny na principu rozpoznávání optického charakteru a na zjišťování vzdálenosti mezi obrazy. Neuronové sítě založené na příznacích jsou trénovány na

segmentu obrázků. Tyto obrázky jsou založeny na rozdílech v textuře, nebo na kombinaci rozdílu textury a tvaru. Dále je možné provádět segmentaci s využitím odhadu vzdáleností, prahování a mapování histogramu, odhadu optického toku, spojování hran a narůstání oblastí.

2.4.4. Rozpoznání objektu

Rozpoznávání objektu se skládá z určení pozice, orientace a velikosti objektu. Hlavním smyslem je také přiřazení značky detekovanému objektu. Na základě poznatků z dostupné literatury se ve většině aplikací neuronových sítí pro rozpoznávání objektů v obrazu, používá princip založený na *základě hodnot pixelů* (based on pixel data). Další metodou, jak rozpoznávat objekty v obrazu, je použití metody *příznaků* (based on features).

K rozpoznávání objektů v obrazu na *základě hodnot pixelů* (based on pixel data) existuje několik typů neuronových sítí, které je možné využít k tomuto účelu. Jedná se například o *dopřednou neuronovou síť, její varianty se sdílenými vahami, rekurzivní síť, ART (Adaptive Resonance Theory) síť a jejich modifikace, evoluční fuzzy neuronové síť, autoasociativní paměti, Neocognitron a Hopfieldovy síť*.

Zásadním problémem procesu rozpoznávání objektů je to, že některé objekty v obrazu mohou mít různou polohu, mohou být různě natočeny a mohou mít různou velikost. Tyto problémy nám mohou zásadním způsobem ovlivnit výsledek v procesu rozpoznávání objektů v obrazu. Za účelem vyrovnat se s těmito změnami, bylo nutné vyvinout několik nových architektur neuronových sítí.

Metoda rozpoznávání objektů v obrazu na *základě příznaků* (based on features) zkoumá lokální vlastnosti objektů na více než jednom pixelu. K tomuto účelu se používají *dopředné neuronové síť, Hopfieldovy síť, fuzzy neuronové síť a RAM neuronové síť*. Tyto neuronové síť se zaměřují během procesu rozpoznávání objektů v obrazu na vývoj a výběr nejlepších vlastností (příznaků). Hlavním určujícím parametrem těchto metod je to, že příznaky jsou nezávislé na změně rotace a velikosti objektu. Během procesu rozpoznávání objektů se zkoumají hlavně lokální geometrické vlastnosti. Jedná se především o vlastnosti, jako jsou *Furierovy deskriptory odvozené z obrazu, transformace pomocí waveletů, hlavní komponenty získané z obrazu, projekce subobrazu na x-ovou a y-ovou osu, Zernike momenty a momenty HU. Furierovy deskriptory, Zernike momenty a momenty HU* jsou invariantní vzhledem k pozici, orientaci a velikosti objektu.

2.4.5. Porozumění obrazu

Porozumění obrazu je velice složitá oblast ve zpracování obrazu. V této části se využívají techniky ze segmentace obrazu a rozpoznávání objektů v obrazu, které mají znalost obsahu obrazu. Pro porozumění obrazu je možné použít dopředné

neuronové sítě. Tento typ neuronové sítě byl použit pro ovládání robota na základě analyzování obrázků z kamery.

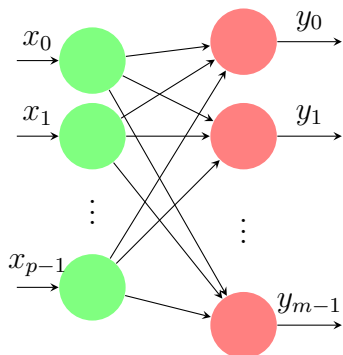
Prvním hlavním problémem neuronových sítí ve fázi porozumění obrazu je fakt, že neuronová síť je chápána jako „černá skříňka“. Neuronové síti poskytneme na vstup celou řadu vstupních dat. Tato síť nám pro každý vstup vrátí daný výstup. Charakter „černé skříňky“ neuronové sítě spočívá v tom, že nevíme jak na tyto výsledky síť přišla a zda jsou tyto výsledky přesné. Druhým hlavním problémem je znalost obsahu vstupních dat.

3. Neuronové sítě realizující PCA a rozpoznávání

Při psaní této kapitoly jsem vycházel převážně z knih [4], [5], [9], [11] a z článků [6], [7] a [13]. Stejně jako člověk i počítač se musí naučit správně rozpoznat a identifikovat známe objekty v daném prostředí na základě vizuální informace. Tuto vizuální informaci je člověk schopen vyextrahovat ze známých a neznámých objektů bez vážnějších problémů. Může to být např. barva auta, tvar objektu, věk nebo pohlaví osoby a další. Oproti tomu strojové rozpoznávání objektů je však daleko složitější. Pro extrakci příznaků ve strojovém učení se nabízí použití celá řada přístupů. V této práci jsem se zaměřil na extrakci příznaků pomocí neuronových sítí. Tato kapitola se zabývá dvěma typy neuronových sítí - GHA a APEX. Námi uvedené neuronové sítě jsou schopné provádět efektivní výpočet *analýzy hlavních komponent* (PCA). Cílem této kapitoly je extrakce příznaků, které nám dobře reprezentují obrazovou informaci dostatečným způsobem tak, aby v další fázi strojového rozpoznávání objektů v obraze mohl klasifikátor na základě vektoru příznaků správně identifikovat daný objekt.

3.1. Neuronová síť GHA

První neuronovou sítí, která je schopna provádět efektivní výpočet *analýzy hlavních komponent* (PCA) je GHA - *zobecněný Hebbův algoritmus*. Jedná se o síť, kde se provádí proces *učení bez učitele*. Schéma této sítě je znázorněno na obrázku č. 6.



Obrázek 6. Dopředná síť s lineárními neurony ve výstupní vrstvě.

Struktura této neuronové sítě je tvořena z p vstupů a z m výstupů. Neurony, které se nacházejí ve výstupní vrstvě, jsou lineární. Pro tento typ neuronové sítě je typické, že má méně výstupů, než vstupů, tj. ($m < p$). Jediným předmětem v procesu učení je úprava synaptických vah w_{ji} , kde $i = 0, 1, \dots, p - 1$ a $j = 0, 1, \dots, m - 1$. Tedy w_{ji} spojuje neuron i ve vstupní vrstvě s neuronem j ve výstupní vrstvě.

Výstup $y_j(n)$ neuronu j v čase n je definován jako odezva na množinu vstupů $\{x_i(n) | i = 0, 1, \dots, p-1\}$, tedy

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji}(n)x_i(n), \text{ pro } j = 0, 1, \dots, m-1.$$

Synaptická váha $w_{ji}(n)$ se upravuje v průběhu fáze učení pomocí *zobecněného Hebbova algoritmu*. Matematicky se dá úprava synaptické váhy popsat jako

$$\Delta w_{ji}(n) = \eta \left(y_j(n)x_i(n) - y_j(n) \sum_{k=0}^j w_{ki}(n)y_k(n) \right), \quad \begin{array}{l} i = 0, 1, \dots, p-1, \\ j = 0, 1, \dots, m-1 \end{array}$$

kde $\Delta w_{ji}(n)$ je úprava váhy $w_{ji}(n)$ v čase n a η označuje rychlost učení neuronové sítě.

Algoritmus pro GHA (Generalized Hebbian Algorithm)

Krok 1: Provede se inicializace v čase $n = 1$ synaptických vah w_{ji} neuronové sítě na malé náhodné hodnoty. Parametr rychlosti učení η se nastaví na malé kladné číslo.

Krok 2: Pro $n = 1, j = 0, 1, \dots, m-1$ a $i = 0, 1, \dots, p-1$ vypočítej

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji}(n)x_i(n),$$

$$\Delta w_{ji}(n) = \eta \left(y_j(n)x_i(n) - y_j(n) \sum_{k=0}^j w_{ki}(n)y_k(n) \right),$$

kde x_i je i -tá složka vstupního vektoru $x(n)$ typu $p \times 1$ a m je počet hlavních komponent.

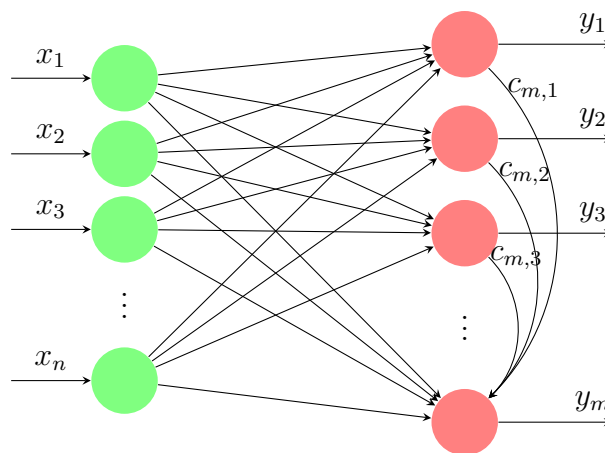
Krok 3: Provedeme inkrementaci n o 1, pokračujeme krokem 2 dokud synaptické váhy w_{ji} nedosáhnou ustálené hodnoty. Pro velké n synaptická váha w_{ji} neuronu j konverguje k i -té složce vlastního vektoru, který je asociován s j -tým vlastním číslem korelační matice vstupního vektoru $x(n)$.

3.2. Neuronová síť APEX

Neuronová síť *GHA* (Generalized Hebbian Algorithm) je založena na dopředných spojeních. V této sekci popíšeme neuronovou síť, která se označuje *APEX* (Adaptive Principal Components Extraction). Zkratka *APEX* znamená adaptivní extrakci hlavních komponent. Tato síť je založena na dopředných a zpětnovazebních spojeních. Tento algoritmus má zajímavou vlastnost. Jestliže máme k dispozici prvních i hlavních komponent, potom tento algoritmus *APEX* ví, jak vypočítat $i + 1$ hlavních komponent iterativně.

Na obrázku č. 7. je znázorněn model neuronové sítě *APEX*. Vstupní vektor má dimenzi n . Složky tohoto vektoru jsou x_1, \dots, x_n . Všechny výstupní neurony jsou lineární. V této neuronové síti *APEX* se využívají dva druhy synaptických spojení:

- *Dopředná spojení* z vstupních neuronů do každého neuronu $1, 2, \dots, m$ ve výstupní vrstvě, kde $m < n$. Tato spojení jsou reprezentovány maticí W dopředným synaptických vah. Dopředná spojení pracují podle *Hebbova pravidla učení* a jsou *excitační*.
- *Laterální spojení* z výstupu jednotlivých neuronů $1, 2, \dots, i-1$ do neuronu i . Tímto se do neuronové sítě zavádí zpětná vazba. Tato spojení jsou reprezentovány maticí C . *Laterální spojení* pracují podle *antihebbovského pravidla učení*. To znamená, že jsou *inhibiční*.



Obrázek 7. Neuronová síť *APEX*.

Výstup y_i i -tého neuronu je dán lineárním vztahem mezi vstupy x_1, \dots, x_n a výstupy y_1, \dots, y_{i-1} neuronů 1 až $i-1$. Konkrétně dostáváme

$$y_i = \sum_{j=1}^n w_{ij}x_j - \sum_{j=1}^{i-1} c_{ij}y_j,$$

$$\mathbf{y} = \mathbf{W}\mathbf{x} - \mathbf{C}\mathbf{y},$$

kde $\mathbf{y} = (y_1, \dots, y_m)^T$ je výstupní vektor, $\mathbf{W} = (w_{ij}), i = 1, \dots, m, j = 1, \dots, n$ je matice dopředných synaptických vah a

$$C = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ c_{2,1} & 0 & \cdots & \cdots & 0 \\ \vdots & & \ddots & & \vdots \\ c_{m-1,1} & \cdots & c_{m-1,m-2} & 0 & 0 \\ c_{m,1} & \cdots & \cdots & c_{m,m-1} & 0 \end{pmatrix}$$

je dolní trojúhelníková matice zpětnovazebních vah. Stejně jako u modelu *GHA*, model *APEX* udržuje hierarchické pořadí výstupních neuronů, protože i -tý výstup závisí na výstupu předchozích $i - 1$ neuronů.

Algoritmus APEX (Adaptive Principal Components Extraction)

Krok 1: Proveďte se inicializace v čase $k = 1$ vektoru dopředných vah w_i a vektoru zpětnovazebních vah c_i neuronové sítě na malé náhodné hodnoty. Parametr rychlosti učení η se nastaví na malé kladné číslo.

Krok 2: Pro $i = 1, \dots, m$ a $j = 1, \dots, n$ vypočítej

$$\begin{aligned} \Delta w_{ij}(k) &= \beta(k)[y_i(k)x_j(k) - y_i(k)^2 w_{ij}(k)], \\ \Delta c_{ij}(k) &= \beta(k)[y_i(k)y_j(k) - y_i(k)^2 c_{ij}(k)], \text{ pro } j < i \end{aligned}$$

Krok 3: Inkrementuj i o 1, přejdi na krok 2 a pokračuj až po $i = m - 1$, kde m se rovná počtu požadovaných hlavních komponent. Pro velké k platí $w_i(k) \rightarrow q_i(k)$ a $c_i(k) \rightarrow 0$, kde q_i je vlastní vektor odpovídající i -tému vlastnímu číslu korelační matice vektoru $x(k)$.

3.3. Popis objektu

V reálném světě využívá člověk pro popis objektů jejich specifických vlastností. Tyto vlastnosti jsou označovány jako příznaky. Tyto příznaky se potom používají ve fázi klasifikace objektů. Příznakem u daného objektu může tak být například barva, velikost a další. Takto získané příznaky musí co nejpřesněji a správně vystihnout charakteristické vlastnosti zkoumaného objektu. Kvalita rozpoznávání objektů tedy závisí na vhodné volbě příznaku. Mezi nejdůležitější vlastnosti příznaků patří *invariantnost* (nezávislost příznaku na změně jasu, kontrastu, translace, rotace a změně měřítko), *spolehlivost* (objekty spadající do stejné třídy mají podobné hodnoty příznaků), *diskriminabilita* (objekty v různých třídách mají různé hodnoty příznaků) a další.

Pro popis objektu rozlišujeme dva základní přístupy – *strukturální popis* a nebo *popis pomocí příznakového vektoru*.

Strukturální popis objektu není předmětem této práce. Metody strukturálního popisu objektu jsou založeny na popisu objektů pomocí základních primitiv a popisu jejich vztahů mezi nimi. Vztahy mezi primitivy se nazývají relace. Tyto

relace mohou být prostorové, funkční a další. Každý obraz je tvořen základními tvary primitiv, jako jsou čtverec, obdélník, kruh a další. Příkladem relací mezi těmito primitivy jsou třeba *dotýkat se*, *protínat se* nebo *být uvnitř*. Spojením těchto relací jsme poté schopni popsat obraz.

Popis objektu *pomocí příznakového vektoru* využijeme v této práci. Cílem tohoto přístupu je matematicky popsat zkoumaný objekt pomocí vektoru X . Vektor X se označuje jako *příznakový vektor* a skládá se z jednotlivých složek x_i . Každá i -tá složka vektoru X bude reprezentovat i -tý příznak. *Příznakový vektor* může mít libovolnou dimenzi (počet příznaků). Každý takový zkoumaný objekt můžeme tedy vyjádřit jako bod v prostoru příznaků, jehož dimenze je stejná, jako dimenze *příznakového vektoru*. Každý přidání příznak x_i do *příznakového vektoru* X tak navíc zvyšuje náročnost výpočtu a prodlužuje dobu výpočtu. Pro omezení celkového počtu příznaků na ty nejdůležitější příznaky, je nutné provést *redukcí dimenze příznakového prostoru*. K tomuto účelu lze využít dva základní přístupy – *extrakce příznaků* nebo *selekce příznaků*.

Extrakce příznaků (feature extraction) – Tento přístup spočívá v provedení transformace z původního p -rozměrného prostoru dat, do prostoru příznaků. Výsledná dimenze prostoru příznaku je m , kde $m < p$. Takto nově získané příznaky se významově liší od původních.

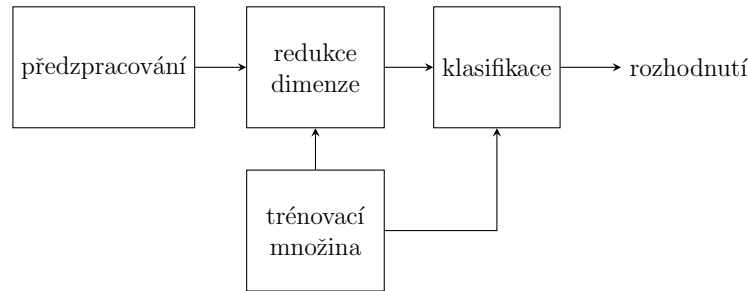
Selekce příznaků – Zde se provádí pouze výběr podmnožiny příznaků z původního prostoru dat bez transformace, které mají maximální diskriminativitu.

3.4. Aplikace rozpoznávání objektů v obraze

K rozpoznávání objektů v obraze využijeme neuronové sítě. Hlavním cílem tohoto procesu je správně identifikovat objekt obsažený v obraze. Trénovací množina je tvořena portrétními fotografiemi různých lidí. Každý systém na rozpoznávání objektů v obraze musí řešit celou řadu problémů týkající se úložiště obrazů, předzpracování obrazu, extrakce příznaků a rovněž i následnou klasifikaci. Rozpoznávání portrétů lidí je velmi obtížné. Obtížnost spočívá v tom, že všechny portrétní fotografie lidí jsou si velice podobné. Například na všech portrétních fotografiích se vyskytují oči, nos, ústa atd. Jde tedy o to, porovnat hledanou portrétní fotografii s portrétními fotografiemi všech lidí v trénovací množině. Tento systém musí tedy správně rozpoznat i neznámé portrétní fotografie, které nejsou součástí trénovací množiny. Na obrázku č. 8. máme znázorněno schéma našeho systému na rozpoznávání portrétních fotografií lidí. Toto schéma vychází z jednotlivých fází popsaných v kapitole 2.4.

Naš systém pro rozpoznávání objektů v obraze se skládá z následujících částí: *předzpracování*, *redukce dimenze*, *trénovací množina* a *klasifikace*.

V části *předzpracování* provádíme hlavně normalizaci velikosti obrazu, která je důležitá pro výkonnost našeho systému. Dále je obraz převeden z barevného obrazu RGB do šedotónového obrazu. O principu, jak převést barevný obraz



Obrázek 8. Schéma systému na rozpoznávání objektů.

RGB do šedotónového obrazu, pojednává podkapitola 2.1.

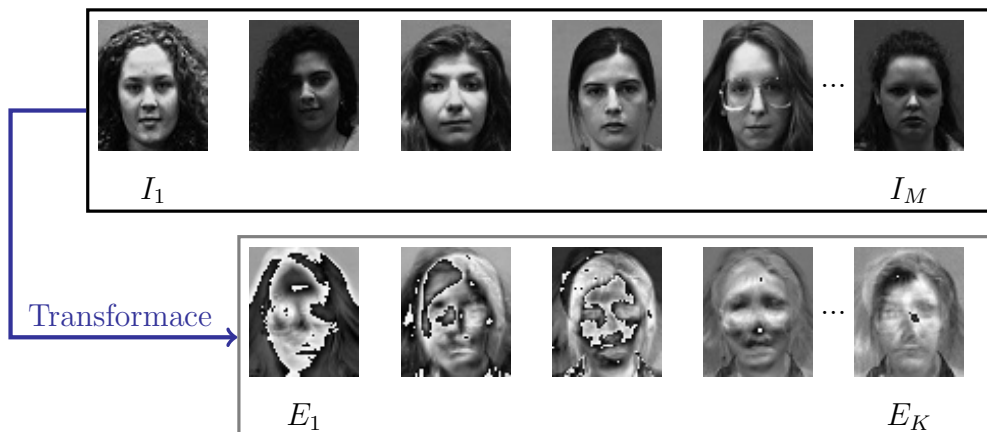
Po provedení části předzpracování máme k dispozici normalizovaný šedotónový obraz. Tento obraz je přiveden na vstup neuronové sítě, která nám zde provádí *redukci dimenze* pomocí extrakce příznaků. Redukce dimenzionality je řešena pomocí neuronových sítí, které jsou představeny v podkapitole 3.1 a 3.2. Takto získáme klíčové příznaky, které se potom využijí v části klasifikace.

V části *klasifikace* se využije vzorový obraz (vzor), který zadal uživatel. Tento vzor je předložen dané neuronové síti GHA nebo APEX. Pro měření podobnosti příznaků se použije vhodné kritérium. My jsme zvolili *Euklidovskou vzdálenost*. Na základě výsledků euklidovské vzdálenosti můžeme vzor klasifikovat.

3.5. Přístup pomocí eigenfaces

Eigenfaces přístup je metoda analýzy hlavních komponent (PCA). Počátky metody analýzy hlavních komponent (PCA) položil v roce 1901 Karl Pearson. Většinou se používá jako nástroj v exploratorní analýze dat a pro vytváření prediktivních modelů (např. rozpoznávání portrétních fotografií). Metoda PCA pracuje v případě rozpoznávání portrétních fotografií s množinou obrazů, které tvoří vysoce dimenzionální dataset. Tyto jednotlivé obrazy jsou mapovány jako body ve vysoce dimenzionálním prostoru. Metoda analýzy hlavních komponent (PCA) je matematická metoda, která používá ortogonální transformaci pro převod množiny M obrazů portrétních fotografií do množiny K nekorelovatelných vlastních vektorů, které se nazývají *eigenfaces*. Tyto vektory potom generují podprostor tváří, který se nazývá *facespace*. Tyto vlastní vektory (eigenfaces) si můžeme představit jako množinu příznaků, které charakterizují rozdíly mezi portrétními fotografiemi. Počet eigenfaces je vždy menší než počet originálních obrazů, tj. $K < M$.

Na obrázku č. 9. je příklad trénovací množiny, která obsahuje celkem M obrazů. Pomocí transformace získáme K eigenfaces pro reprezentaci trénovací množiny, které jsou znázorněny na obrázku č. 11. Tato transformace je definována tak, že první eigenface ukazuje nejvýraznější příznaky trénovací množiny obrazů. Každý další eigenface v pořadí zobrazuje další nejvíce výrazné příznaky s tím, že

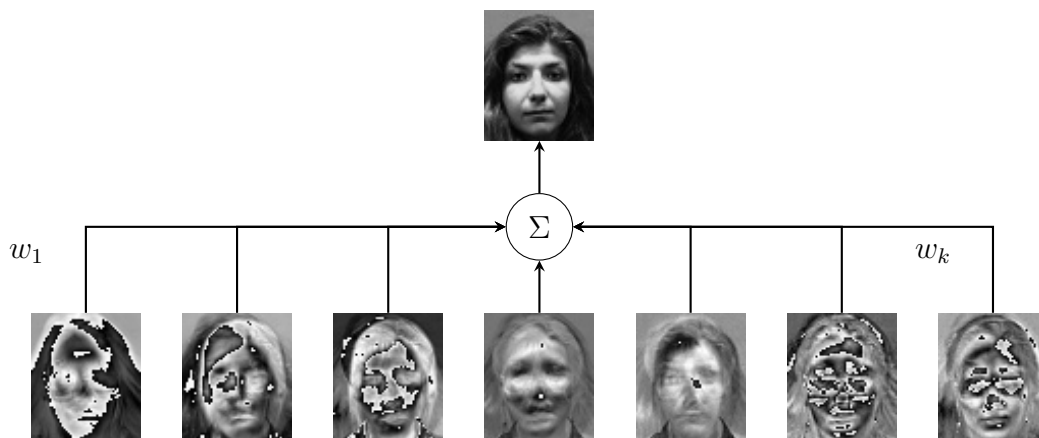


Obrázek 9. Transformace trénovací množiny do K vybraných eigenfaces.

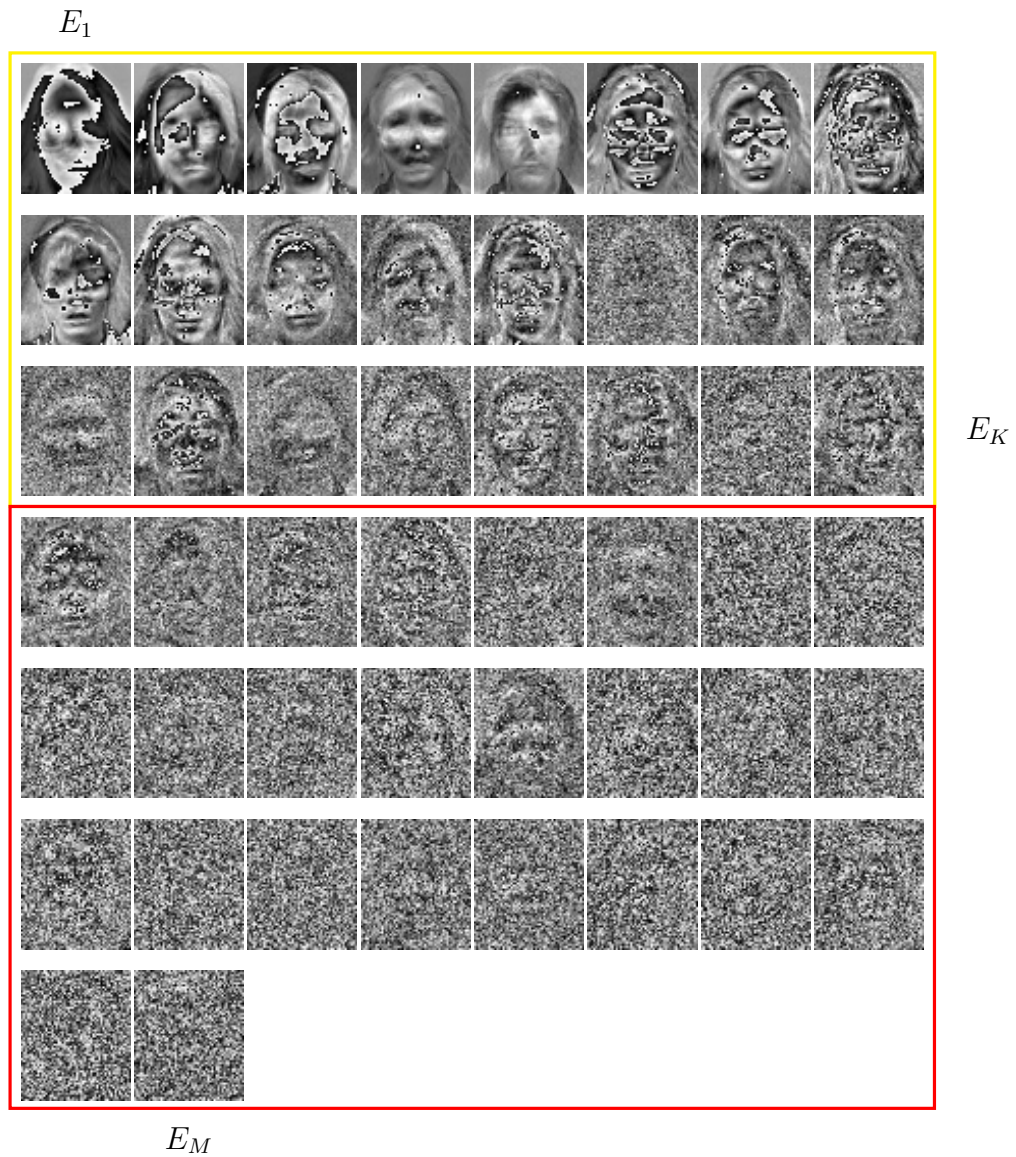
musí být nekorelovatelný s předcházejícím eigenfacem. Navíc každý další eigenface v pořadí ukazuje menší příznaky a více šumu. Pouze několik málo prvních eigenfaces (řekněme K) jsou vybrány. Na obrázku č. 11 je těchto K eigenfaces znázorněno ve žlutém rámečku. Zbytek posledních eigenfaces, které jsou označeny v červeném rámečku, jsou vyřazeny.

Těchto K eigenfaces mohou bezpečně reprezentovat celou původní trénovací množinu, protože zobrazují hlavní příznaky, které tvoří dataset. Každý obraz v originální trénovací množině může být reprezentován z hlediska těchto K eigenfaces.

Každá tvář může být přesně reprezentována jako lineární kombinace všech těchto eigenfaces. Každou tvář můžeme také reprezentovat pouze pomocí určitého počtu K nejlepších eigenfaces. Ty nejlepší eigenfaces jsou ty, které mají největší vlastní čísla. Nejlepších K eigenfaces tvoří M -dimenzionální podprostor, který nazýváme facespace všech možných obrazů. Je to zobrazeno na obrázku č. 10.



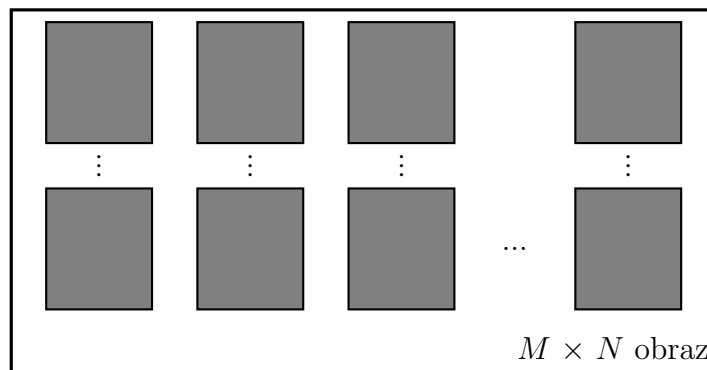
Obrázek 10. Reprezentace tváře pomocí lineární kombinace eigenfaces.



Obrázek 11. Vypočítané eigenfaces z dané trénovací množiny.

3.6. Výpočet eigenfaces

Metoda PCA pomocí eigenfaces považuje každý pixel v obraze jako samostatnou dimenzi. Jednotlivé obrazy portrétních fotografií, které jsou obsaženy v trénovací množině budou reprezentovány dvoudimenzionálním polem velikosti $M \times N$. Toto pole obsahuje hodnoty z rozsahu 0 až 255. Všechny obrazy portrétních fotografií musí být přesně stejné velikosti, které budeme reprezentovat vektorem dimenze $M \times N$. Taková trénovací množina je potom zobrazena na obrázku č. 12.



Obrázek 12. Trénovací množina, která obsahuje všechny obrazy stejné dimenze.

Každý obraz o rozměrech $M \times N$ z trénovací množiny budeme tedy reprezentovat jako sloupcový vektor dimenze $M \times N$ nebo si daný obraz můžeme představit jako bod v $(M \times N)$ -dimenzionálním prostoru. Potom tedy dochází k mapování celé kolekce obrazů v trénovací množině na body tohoto vysoce dimenzionálního prostoru.

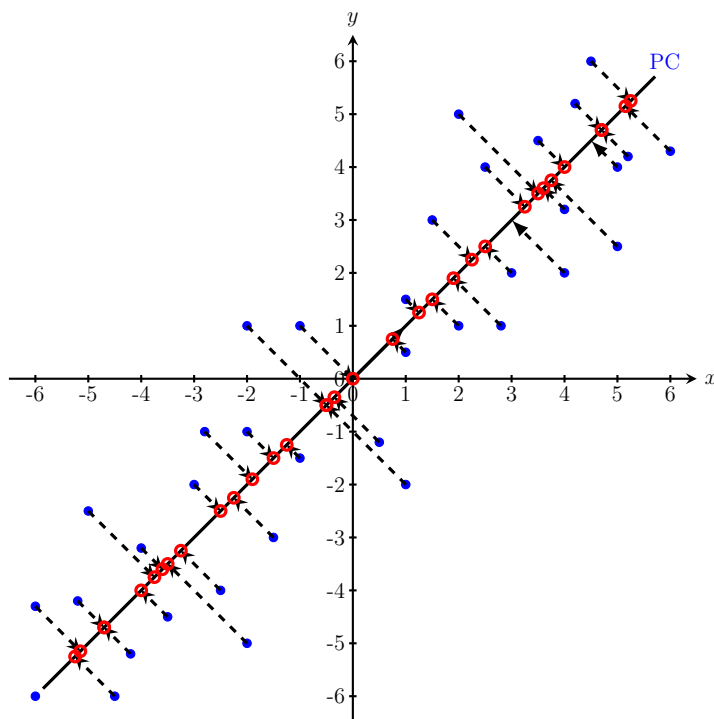
Obrazy portrétních fotografií jsou si podobné ve více či méně rysech. Z toho plyne, že tyto obrazy portrétních fotografií nebudou v tomto vysoce dimenzionálním prostoru rozdistributedovány náhodně a můžou tak být popsány podprostorem s mnohem nižší dimenzí. Pomocí metody analýzy hlavních komponent (PCA) nalezneme tedy vlastní vektory (eigenfaces), které nejlépe popisují distribuci obrazů portrétních fotografií v celém prostoru obrazů. Tyto vektory definují podprostor obrazů portrétních fotografií, který se nazývá facespace. Každý vektor v tomto podprostoru má tedy délku $M \times N$ a popisuje tak obraz o rozměrech $M \times N$.

Pro výpočet vlastních vektorů (eigenfaces) a provedení redukce dimenzionality vstupních dat jsme využili neuronové sítě realizující výpočet PCA (Principal Component Analysis). Konkrétně se jednalo o neuronové sítě *GHA* (Generalized Hebbian Algorithm) a *APEX* (Adaptive Principal Components Extraction). Pro příklad jsou na obrázku č. 11. zobrazeny eigenfaces, které byly vypočteny pomocí sítě GHA.

Je zřejmé, že v případě obrázků z trénovací množiny se bude jednat o $M \times N$ dimenzionální prostor. Nejprve budou tyto obrazy postupně předloženy neuronové síti, která vypočte podprostor obrazů facespace. Pro jednodušší představu uvažujeme obrazy v trénovací množině o rozměrech 1×2 . Tyto původní obrazy lze reprezentovat pomocí jediné hlavní komponenty PC (černá čára). Potom daná neuronová síť vygeneruje podprostor facespace, který je znázorněn na obrázku č. 13. Modré body reprezentují původní obrazy trénovací množiny. Červené body znázorňují projekci daného obrazu do podprostoru facespace. Tyto červené body představují nové hodnoty v novém souřadnicovém systému. V našem případě 2-dimenzionální data reprezentujeme pomocí 1-dimenzionálních. Na výstupu neu-

ronové sítě jsou vektory, které generují podprostor facespace.

V naší aplikaci pracujeme s trénovací množinou, která obsahuje obrazy portrétních fotografií o rozměrech 54×64 . Potom má výsledný prostor dimenzi 3456. Takto vysoce dimenzionální prostor redukuje pomocí dané neuronové sítě na podprostor s nižší dimenzí.

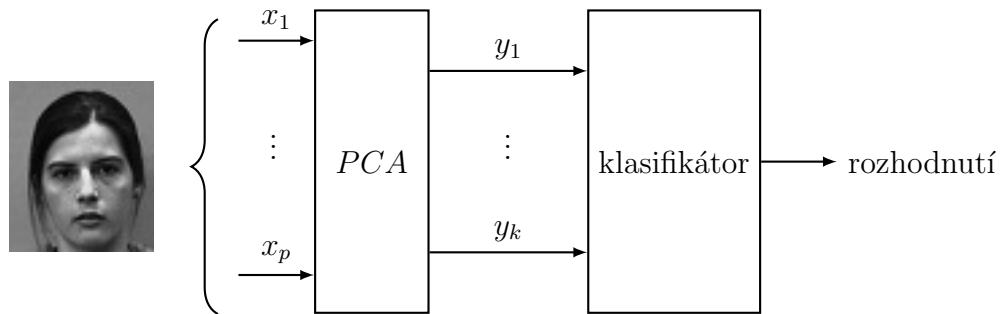


Obrázek 13. PCA pro obrazy o rozměrech 1×2 .

3.7. Použití eigenfaces pro klasifikaci tváří

Rozpoznávání vzorů je fáze, která výrazně závisí na reprezentaci vstupních dat. Hlavním cílem rozpoznávání vzorů je rozhodnout, do jaké třídy spadá zkoumaný vzor. Pro popis každého obrazu jsme použili metodu *extrakce příznaků* (*feature extraction*). Tato metoda byla realizována pomocí neuronových sítí, kterými jsme se zabývali v podkapitole č. 3.1 a podkapitole č. 3.2. Vycházíme z toho, že máme k dispozici trénovací množinu, která obsahuje celkem M obrazů. Dále máme k dispozici neznámý obrázek. Všechny tyto obrazy jsou stejné velikosti, tj. $M \times N$. Na této trénovací množině jsme danou síť natrénovali. Tímto krokem jsme tedy provedli fázi učení dané neuronové sítě a získali jsme tak matici vah. V případě rozpoznávání vzoru pomocí příznaků je možné tuto úlohu znázornit pomocí schématu uvedeném na obrázku č. 14.

Potom ve fázi rozpoznávání daná neuronová síť provede pro každou portrétní fotografii trénovací množiny projekci této fotografie do vlastních vektorů (eigen-



Obrázek 14. Schéma pro rozpoznávání tváří pomocí PCA.

faces), které generují podprostor facespace a byly získány ve fázi natrénování sítě. Pomocí této projekce získáme reprezentaci každé portrétní fotografie ve formě K nejdůležitějších hlavních komponent.

Při rozpoznávání neznámé portrétní fotografie, která není součástí trénovací množiny se postupuje následovně. Neznámá portrétní fotografie má stejný rozměr jako mají obrazy v trénovací množině. Tuto neznámou portrétní fotografii převedeme nejprve na sloupcový vektor (x_1, \dots, x_p) a ten přivedeme na vstup neuronové sítě, která realizuje PCA. Ta provede projekci naší neznámé portrétní fotografie do vlastních vektorů (eigenfaces). Pomocí této projekce získáme reprezentaci neznámé portrétní fotografie ve formě K nejdůležitějších hlavních komponent. Takto získané hlavní komponenty naší neznámé fotografie (y_1, \dots, y_k) se potom porovnají se všemi hlavními komponentami trénovacích portrétních fotografií.

Na základě vhodného kritéria je možné rozhodnout zda je portrétní fotografie známá nebo neznámá. V případě, že je portrétní fotografie známá, můžeme rozhodnout o její identitě. Jako již zmiňované kritérium můžeme použít např. *Euklidovskou vzdálenost*.

Euklidovská vzdálenost (Euclidean distance) je nejvíce používanou metodou pro měření vzdálenosti kvantitativních proměnných. Představuje měření vzdálenosti dvou bodů tak, jako by byla změřena pravítkem – jde tedy o přímou vzdálenost mezi dvěma body. Vypočte se jako

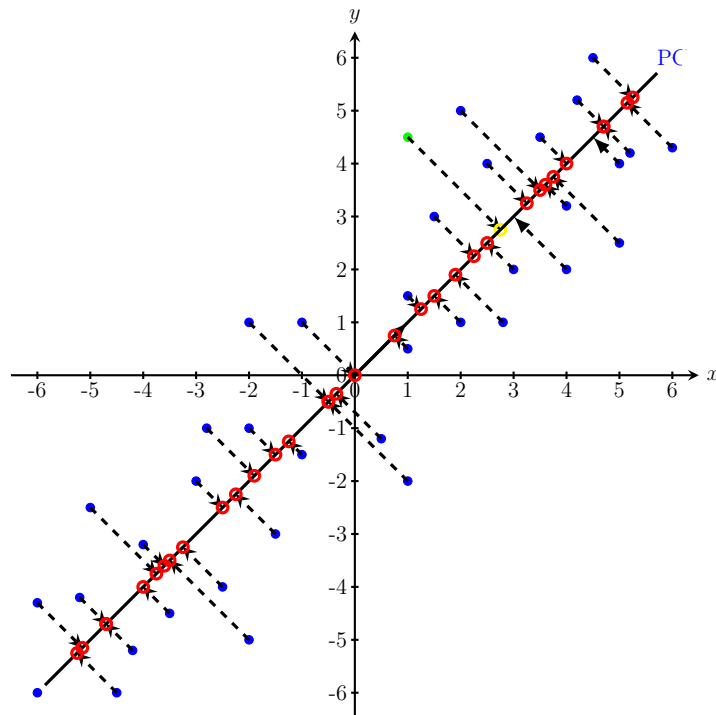
$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2},$$

kde d_{ij} vyjadřuje vzdálenost (odlišnost) bodu i od bodu j . Dále x_{ik} je k -tý příznak objektu i a x_{jk} je k -tý příznak objektu j a n je počet příznaků popisující objekt.

Tato vzdálenost je určena jako druhá odmocnina sumy čtvercových vzdáleností mezi souřadnicemi objektů. Je vždy větší, nebo rovna nule (pro identické body), vyšší hodnoty vychází pro body vykazující menší podobnost. Existuje také několik druhů této metody, např. Čtvercová Euklidovská vzdálenost

$$d_{ij} = \sum_{k=1}^n (x_{ik} - x_{jk})^2.$$

Pro jednodušší představu opět uvažujeme obrazy v trénovací množině o rozměrech 1×2 . Tyto původní obrazy lze reprezentovat pomocí jediné hlavní komponenty PC (černá čára). Potom daná neuronová síť vygeneruje podprostor facespace, který je znázorněn na obrázku č. 15. Modré body reprezentují původní obrazy trénovací množiny. Červené body znázorňují projekci daného obrazu do podprostoru facespace. Tyto červené body představují nové hodnoty v novém souřadnicovém systému. V našem případě 2-dimensionální data reprezentujeme pomocí 1-dimensionálních. Na výstupu neuronové sítě jsou vektory, které generují podprostor facespace. Neznámý obraz o rozměrech 1×2 , který nebyl součástí trénovací množiny, je na obrázku č. 15. znázorněn zeleným bodem. Jeho projekce do podprostoru facespace je znázorněna žlutým bodem. Poté, co jsme provedli projekci, můžeme přistoupit k rozhodnutí, zda daný obraz je známý nebo neznámý. K tomu využijeme již zmíněnou euklidovskou vzdálenost žlutého bodu od všech ostatních červených bodů.



Obrázek 15. Aplikace euklidovské vzdálenosti pro rozpoznávání neznámého obrazu (zelený bod).

4. Uživatelská příručka

4.1. Popis aplikace

Cílem práce bude aplikace, rozpoznávající objekty v obraze. Tato aplikace bude využívat pro redukci dat dva různé typy neuronových sítí. Jsou to neuronové sítě *GHA* a *APEX*. Jako objekty jsme zvolili portrétové fotografie.

Výsledná aplikace byla naprogramována v programovacím jazyce *Microsoft Visual C#*. Architektura této aplikace je založena na platformě *Microsoft .NET Framework 4.5*. Jako vývojové prostředí bylo zvoleno *Microsoft Visual Studio 2013*.

4.2. Požadavky na instalaci

Aplikaci je nutné spouštět na počítačích s operačním systémem Windows 7 a vyšší. Dále je nutnou podmínkou mít na počítači nainstalovaný Microsoft .NET Framework 4.5 nebo vyšší.

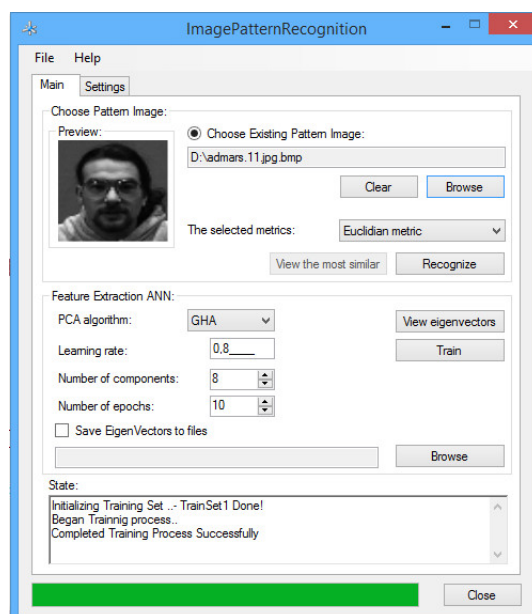
4.3. Hlavní okno aplikace

Aplikace je tvořena jedním hlavním oknem a několika dílčími okny. Hlavní okno aplikace je znázorněno na obrázku č. 16. V hlavním okně se nachází hlavní menu a dvě záložky – *Main* a *Settings*.

4.4. Záložka Main

První záložka *Main* tvoří jádro celého programu. Tato záložka je rozčleněna do tří sekcí. Sekce *Feature extraction ANN* se týká fáze redukce dat pomocí neuronových sítí. Ve fázi redukce dat se nachází mnoho nastavitelných parametrů, které vstupují do výpočtu neuronových sítí. V této sekci máme možnost zvolit u položky *PCA algorithm* typ dané neuronové sítě, která bude realizovat redukci dimenzionality vstupních dat. Na výběr máme ze dvou neuronových sítí – *GHA* a *APEX*. Dalším významným parametrem je položka *Learning rate*, která ovlivňuje rychlost učení vybrané neuronové sítě. Parametr *Number of components* má zcela zásadní vliv na fázi redukce dimenzionality dat. Pomocí tohoto parametru jsme schopni ovlivnit velikost dimenze původních vstupních dat. Posledním parametrem v této sekci je *Number of epochs*. Tento parametr nastavuje počet epoch pro danou neuronovou síť. Udává, kolikrát bude celá trénovací množina předložena zvolené neuronové síti. V případě, že jsme provedli na záložce *Settings* výběr a načtení trénovací množiny, můžeme po nastavení parametrů v sekci *Feature extraction ANN* zahájit trénování zvolené neuronové sítě. K tomu

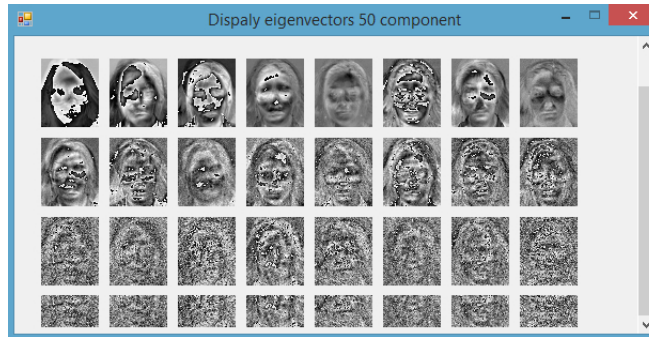
slouží tlačítko *Train*. Průběh trénování dané neuronové sítě je znázorněn na progressbaru, který se nachází na spodní straně hlavního okna. Když je daná neuronová síť natrénována na námi zvolené trénovací množině, zpřístupní se tlačítko *View eigenvectors*. Po stisku tohoto tlačítka se otevře okno, které je zobrazeno na obrázku č. 17.



Obrázek 16. Hlavní okno aplikace - záložka Main.

Sekce *Choose pattern image* se týká fáze rozpoznávání a klasifikace. Když je trénování zvolené neuronové sítě úspěšně ukončeno, zpřístupní se důležité prvky této sekce. Především zde uživatel může nahrát neznámý obrázek, který se rozhodl rozpoznat. Máme zde k dispozici tlačítko *Browse*. Pomocí tohoto tlačítka se nahraje vybraný neznámý obrázek do aplikace. Tlačítko *Clear* slouží pro zrušení výběru neznámého obrázku. Posledním tlačítkem této sekce je tlačítko *Recognize*. Po stisku tohoto tlačítka se provede klasifikace neznámého obrázku, který byl nahrán na vstup. Vlevo od těchto tlačítek se nachází náhled nahraného neznámého obrázku v odstínech šedi. Poslední důležitou částí této sekce je parametr *The selected metric*. Tímto parametrem si zvolíme typ pro měření podobnosti objektů ve fázi rozpoznávání a klasifikace. Implicitně je zde na výběr pouze euklidovská vzdálenost.

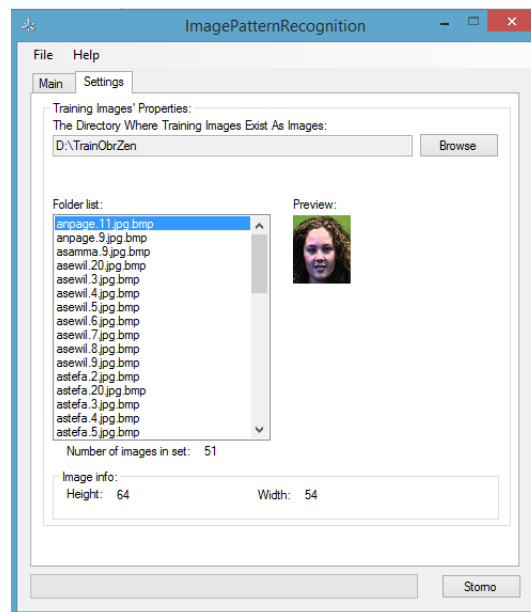
Poslední sekce na této záložce *Main* se nachází ve spodní části okna aplikace. Jedná se o sekci *State*. Tato sekce slouží pro poskytování informací a zobrazování průběhu výpočtu uživateli. Jedná se zejména o zobrazení informací a o zobrazení průběhu nahrávání trénovací sady do aplikace. Tato sekce poskytuje informace o průběhu výpočtu redukce dimenzionality neuronové sítě nad zadanou trénovací množinou.



Obrázek 17. Okno, které zobrazuje eigenvectors.

4.5. Záložka Settings

Druhá záložka v hlavním okně se jmenuje *Settings*. Tato záložka obsahuje pouze tlačítko *Browse*. To slouží pouze pro načtení a následné zpracování trénovací množiny obrázků, umístěné na disku v počítači. Průběh zpracování této množiny je znázorněn v progressbaru. Jednotlivé obrázky trénovací množiny jsou načteny do seznamu *Folder list*. Ten umožňuje procházet jednotlivé obrázky, které jsou součástí trénovací množiny a zobrazit ke každému *Preview*. Pod tímto seznamem *Folder list* je informativní údaj *Number of images in set*, který udává celkový počet obrázků v trénovací množině. Poslední sekci na této záložce je *Image Info*. Zde se pro vybraný obrázek z *Folder list* zobrazí informace o jeho rozměrech.



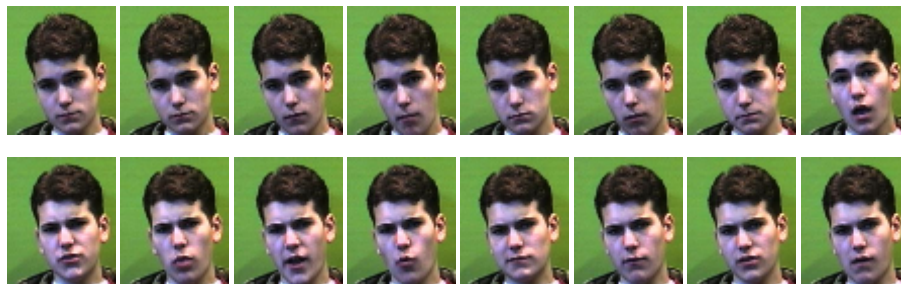
Obrázek 18. Hlavní okno aplikace - záložka Settings.

5. Testy

5.1. Popis vstupních dat

Pro naše experimenty použijeme databázi lidských tváří, která byla získána ze zdroje [16]. Takto získané portrétové fotografie různých lidí budeme používat ve fázi trénování a rozpoznávání námi zkoumaných neuronových sítí *APEX* a *GHA*. Portrétové fotografie byly pořizovány tak, že všechny osoby se nacházely přibližně ve stejné vzdálenosti od fotoaparátu. Následně bylo vytvořeno 20 snímků od každé osoby. Jednotlivé snímky jsou označeny podle následujícího schématu – *jméno.číslosnímku.přípona*, kde *jméno* označuje identitu dané osoby, *číslosnímku* signalizuje, v jakém pořadí byl snímek vytvořen a *přípona* označuje grafický formát obrázku. Během pořizování snímků byl s osobou veden dialog, jehož účelem bylo vyvolat drobné změny ve výrazu obličeje při fotografování každé osoby. Všechny portrétové fotografie, umístěné v této sadě, mají jednotný rozměr 54×64 pixelů. Jsou zde zastoupeny jak fotografie mužů tak i ženy. Na obrázku č. 19. je pro názornost uveden příklad portrétových fotografií konkrétní osoby této množiny.

Tato sada splňuje následující vlastnosti. Každá osoba této množiny byla focena na zeleném pozadí. Všechny fotografie jsou zhotoveny ve stejném měřítku s minimálními změnami otáčení, sklonu a naklonění hlavy. Tyto fotografie byly pořizovány za stejného nasvícení. Na fotografiích jsou rovněž zaznamenány i změny ve výrazu obličeje. Protože jednotlivé snímky byly vytvořeny v daném okamžiku, nejsou tedy k dispozici různé variace účesů.



Obrázek 19. Příklad portrétových fotografií jediné konkrétní osoby.

5.2. Příprava experimentů s neuronovými sítěmi GHA a APEX

Veškeré experimenty s neuronovými sítěmi byly provedeny na počítači Intel Core 2 Duo E8400 3GHz, 4GB RAM. Pro všechny prováděné experimenty jsme si vstupní data rozdělili na 3 základní množiny. První množina je označována

jako trénovací (*TrainSet*). Tato množina obsahuje celkem 109 portrétových fotografií. V této množině jsou všechny portrétové fotografie označeny dle schématu – *jméno.1.přípona*. Portrétové fotografie s tímto označením byly pořízeny jako úplně první fotografie konkrétní osoby v přímém pohledu. Další množina má označení jako testovací sada 1 *TestSet1*. Tato množina obsahuje celkem 506 portrétových fotografií. K označení portrétových fotografií je použito schématu – *jméno.číslosnímku.přípona*, kde jednotlivé snímky jsou číslovány čísly 1 až 20. Je zřejmé, že tato množina obsahuje i obrázky z trénovací množiny *TrainSet*. Tato množina nám slouží pro fázi rozpoznávání. Poslední množinou je testovací sada 2 *TestSet2*. V této množině jsou obrázky označeny dle schématu – *jméno.číslosnímku.přípona*, kde číslo snímku je v rozsahu 11 až 20. *TestSet1* tedy neobsahuje fotografie, obsažené v množině *TrainSet*. Je zde celkem 268 fotografií. Tato množina se také používá pro fázi rozpoznávání. Pro klasifikaci byla využita ve všech experimentech *Euklidovská vzdálenost*.

Při provádění několika různých experimentů jsme získané výsledky shrnuli do následujících tabulek. Struktura tabulek je pro všechny prováděné experimenty stejná a proto zde provedeme vysvětlení významu jednotlivých sloupců tabulek.

- První sloupec tabulky udává typ zvolené neuronové sítě včetně nastavení jejich parametrů. Je to *Rychlost učení (LR)*, *počet hlavních komponent (CC)* a *Počet epoch (CE)*. V každém řádku tohoto sloupce je uveden rozměr obrázků (54×64) a dále jsou uvedeny i nastavené parametry sítě ve tvaru *LR – CC – CE*.
- Druhý sloupec tabulky udává celkový čas, potřebný pro trénování neuronové sítě nad zvolenou trénovací množinou *TrainSet*. Čas je uveden v následujícím formátu – minuty : sekundy . milisekundy.
- Třetí sloupec tabulky se týká první testovací množiny *TestSet1*. Ve sloupci je uveden název testovací množiny a pod ním v závorce je uveden celkový počet fotografií v této množině. V každém řádku tohoto sloupce jsou uvedeny dva údaje – procentní úspěšnost rozpoznání pro množinu *TestSet1* a pod ním je uveden v závorce počet úspěšně rozpoznávaných tváří.
- Čtvrtý sloupec tabulky udává celkový čas potřebný pro rozpoznání neuronové sítě nad testovací množinou *TestSet1*. V tomto sloupci se nachází údaj – celková doba rozpoznání nad množinou *TestSet1*. Čas je uveden v následujícím formátu - minuty : sekundy . milisekundy.
- Pátý sloupec tabulky se týká druhé testovací množiny *TestSet2*. Ve sloupci je uveden název testovací množiny a pod ním v závorce je uveden celkový počet fotografií v této množině. V každém řádku tohoto sloupce jsou uvedeny dva údaje – procentní úspěšnost rozpoznání pro množinu *TestSet2* a pod ním je uveden v závorce počet úspěšně rozpoznávaných tváří.

- Šestý sloupec tabulky udává celkový čas potřebný pro rozpoznání neuronové sítě nad testovací množinou *TestSet2*. V tomto sloupci se nachází údaj – celková doba rozpoznání nad množinou *TestSet2*. Čas je uveden v následujícím formátu - minuty : sekundy . milisekundy.

5.3. Klasifikace v závislosti na změně počtu hlavních komponent sítě

V této části testu budeme sledovat závislost času učení neuronové sítě a úspěšnost klasifikace na změně parametru, který udává počet hlavních komponent. Budeme používat fotografie s rozměrem 54×64 pixelů. Abychom mohli porovnat tyto dva typy neuronových sítí *GHA* a *APEX*, provedli jsme celkem 5 měření pro každý typ neuronové sítě. Pro každou neuronovou síť jsme v tomto měření zafixovali dva parametry (rychlost učení a počet epoch) a sledovali jsme, jaký to bude mít vliv na celkovou dobu trénování sítě a na procentní úspěšnost klasifikace. Při měření jsme měnili parametr počet hlavních komponent u obou sítí. Neuronové sítě *GHA* a *APEX* byly natrénovány nad stejnou trénovací množinou *TrainSet*. Dále tyto sítě byly postupně testovány na dvou testovacích množinách, které byly těmito sítím postupně přikládány na vstup. Je zřejmé, že různě volené hodnoty zafixovaných parametrů ovlivňují úspěšnost klasifikace.

Výsledky testů jsou shrnuty pro neuronovou síť *GHA* v tabulce č. 1. a pro neuronovou síť *APEX* v tabulce č. 2. Z uvedeného je vidět, že pro zafixované parametry rychlost učení 0.1, počet epoch 3 a pro variabilní počet hlavních komponent 68, síť *GHA* dosahuje nejlepších úspěšností v klasifikaci provedené nad množinou *TestSet1* a *TestSet2*. Celková doba pro natrénování této sítě při dané konfiguraci je celkem 5 minut. Z tabulky č. 1. je zřejmé, že při pevně stanovených hodnotách parametrů a při zvětšování hodnoty, odpovídající počtu hlavních komponent, se prodlužuje doba trénování, ale procentní úspěšnost klasifikace se výrazně nemění.

Oproti tomu neuronová síť *APEX* při stejném nastavení jako síť *GHA* vykazuje menší odchylky v procentní úspěšnosti klasifikace. To je způsobeno aktuálním nastavením zafixovaných parametrů. Zásadní rozdíl oproti síti *GHA* je celková doba trénování. Ta je u sítě *GHA* při počtu hlavních komponent 68 celkově 5 minut 6 sekund. Oproti tomu u sítě *APEX* došlo k zásadnímu snížení celkové doby trénování na 48 sekund při stejné konfiguraci sítě, jakou měla síť *GHA*. Z uvedených výsledků měření (tabulka č. 2.) vyplývá, že další postupné zvyšování hodnoty parametru počtu hlavních komponent od hodnoty 68, vede k poklesu úspěšnosti klasifikace. Z uvedeného je vidět, že neuronová síť *APEX* vykazuje při takto pevně nastavených parametrech (rychlost učení sítě a počet epoch) rychlejší proces trénování.

Síť GHA	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.1-1-3	00:00.64	63% (319)	00:01.18	54% (141)	00:00.96
54 × 64 0.1-32-3	01:10.27	87% (443)	00:17.84	81% (210)	00:09.94
54 × 64 0.1-68-3	05:06.20	94% (477)	00:40.33	93% (242)	00:20.76
54 × 64 0.1-75-3	05:58.82	93% (474)	00:46.11	93% (242)	00:22.99
54 × 64 0.1-80-3	07:21.83	93% (473)	00:49.26	91% (235)	00:23.85

Tabulka 1.: Výsledky rozpoznávání pro síť GHA s varia-
bilním parametrem počet komponent.

Síť APEX	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.1-1-3	00:00.74	62% (315)	00:00.58	60% (157)	00:00.28
54 × 64 0.1-32-3	00:22.74	76% (386)	00:16.72	61% (158)	00:08.42
54 × 64 0.1-68-3	00:48.22	77% (390)	00:40.30	68% (176)	00:18.96
54 × 64 0.1-75-3	00:55.27	74% (375)	00:42.34	65% (169)	00:21.37
54 × 64 0.1-80-3	00:57.77	74% (379)	00:44.57	62% (160)	00:23.10

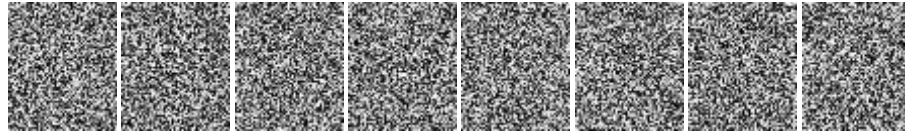
Tabulka 2.: Výsledky rozpoznávání pro síť APEX s vari-
abilním parametrem počet komponent.

5.4. Klasifikace v závislosti na změně rychlosti učení sítě

V této části se zaměříme na výsledky klasifikace v závislosti na změně hodnoty parametru, který udává rychlost učení neuronové sítě. Zde si všimneme také celkové doby trénování neuronové sítě, která se bude odvíjet od nastavené hodnoty pro rychlost učení sítě. Pro naše měření budeme opět používat fotografie, které mají rozměry 54 × 64 pixelů. Pro srovnání těchto dvou typů neuronových sítí *GHA* a *APEX* jsme provedli pro každý typ neuronové sítě celkem 3 měření. Abychom zjistili, jak změna parametru rychlosti učení daného typu neuronové sítě ovlivní úspěšnost klasifikace, bylo nutné zafixovat dva zbývající parametry (počet

komponent a počet epoch). Uvedené dva typy neuronových sítí byly natrénovány nad stejnou trénovací množinou. Testování těchto neuronových sítí proběhlo nad dvěma testovacími množinami (TestSet1 a TestSet2). Je důležité si uvědomit, že uvedené výsledky našich měření byly ovlivněny pevně nastavenými parametry. Je zřejmé, že změna těchto pevně nastavených parametrů ovlivní celkové chování neuronové sítě.

Výsledky klasifikace ovlivněné změnou parametru rychlosti učení jsou shrnuty pro neuronovou síť *GHA* v tabulce č. 3. a pro neuronovou síť *APEX* v tabulce č. 4. Pro potřeby měření jsme stanovili hodnoty počtu hlavních komponent na 8 a počet epoch na 10. Z tabulky č. 3. pro síť *GHA* je zřejmé, že z této konfigurace sítě lze odvodit následující vlastnosti. Z prvního řádku tabulky č. 3. s nastaveným parametrem rychlost učení sítě na hodnotu 0.00008 je vidět, že úspěšnost klasifikace byla 79% pro testovací množinu TestSet1. Pro množinu TestSet2 byla 70%. Pro hodnoty parametrů rychlost učení 0.00008, počet hlavních komponent 8 a počet epoch 10, jsou váhové vektory zobrazeny na obrázku č. 20. Se zvyšující se hodnotou rychlosti učení sítě je vidět z tabulky č. 3., že se zvyšuje procentní úspěšnost klasifikace. Pro hodnotu 0.8 rychlosti učení sítě, síť vykazuje nejlepší úspěšnost klasifikace ze všech 3 našich měření pro testovací množinu TestSet1 a TestSet2. Pro hodnoty parametrů rychlost učení 0.8, počet hlavních komponent 8 a počet epoch 10, jsou váhové vektory zobrazeny na obrázku č. 21. Z uvedeného je zřejmé, že změna hodnoty rychlosti učení sítě se neprojeví na celkové době trénování této sítě. Všechny 3 měření, které jsme pro danou konfiguraci sítě provedli, vykazují celkovou dobu trénování sítě 26 sekund. Z uvedeného plyne, že změna rychlosti učení sítě má vliv na váhové vektory. To je zřejmé i z obrázků č. 20. a č. 21. Dále z toho plyne, že parametr rychlost učení sítě nemá žádný zásadní vliv na celkovou dobu trénování sítě. Je jasné, že hodnotu rychlosti učení sítě nelze zvětšovat do nekonečna. V řádku č. 3 tabulky č. 3. je hodnota rychlosti učení sítě 0.8, pro kterou síť dává nejlepší procentní úspěšnost klasifikace. Dalším pozorováním při našem měření bylo zjištěno, že pokud bychom hodnotu tohoto parametru i nadále zvětšovali, tak síť nebude vykazovat lepší chování a procentní úspěšnost rozpoznávání se výrazným způsobem měnit nebude.



Obrázek 20. Vizualizace váhových vektorů pro síť GHA 0.0008 – 8 – 10.



Obrázek 21. Vizualizace váhových vektorů pro síť GHA 0.8 – 8 – 10.

Síť GHA	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.00008-8-10	00:26.54	79% (400)	00:04.17	70% (182)	00:02.15
54 × 64 0.0008-8-10	00:26.10	84% (428)	00:04.28	79% (204)	00:02.19
54 × 64 0.8-8-10	00:26.50	98% (499)	00:04.33	96% (250)	00:02.10

Tabulka 3.: Výsledky rozpoznávání pro síť GHA s variabilním parametrem rychlost učení sítě.

Síť APEX	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.00008-8-10	00:18.64	85% (434)	00:04.45	80% (228)	00:02.24
54 × 64 0.0008-8-10	00:18.70	85% (434)	00:04.44	71% (184)	00:02.35
54 × 64 0.8-8-10	00:18.87	66% (334)	00:04.25	53% (139)	00:02.16

Tabulka 4.: Výsledky rozpoznávání pro síť APEX s variabilním parametrem rychlost učení sítě.

Z tabulky č. 4. pro neuronovou síť *APEX* je vidět, že 3 řádek odpovídá konfiguraci neuronové sítě (počet hlavních komponent 8, počet epoch 10 a rychlost učení sítě 0.8). Při této konfiguraci sítě bylo dosaženo pro testovací množinu TestSet1 66% úspěšnosti klasifikace a pro množinu TestSet2 53% úspěšnosti kla-

sifikace. To je způsobeno zvýšenou hodnotou rychlosti učení sítě. Dále je z tabulky č. 4. vidět, že když se parametru rychlosti učení přiřadí dostatečně malá hodnota, která zaručuje pomalou úpravu váhových vektorů, potom se zlepšuje procentní úspěšnost klasifikace. Této situaci odpovídá první řádek tabulky č. 4., ve kterém procentní úspěšnost klasifikace dosahuje hodnoty 85% pro množinu TestSet1 a 80% pro množinu TestSet2. Při změně hodnoty rychlosti učení sítě se celková doba trénování sítě výrazným způsobem nemění.

5.5. Klasifikace v závislosti na změně počtu epoch

V této části se zaměříme na výsledky klasifikace v závislosti na změně hodnoty parametru, který udává počet epoch. Parametr počtu epoch udává, kolikrát bude celá trénovací množina TrainSet předložena zvolené neuronové síti. Zde si všimneme závislosti času učení neuronové sítě a úspěšnosti klasifikace na změně parametru, udávajícího počet epoch. Pro naše měření budeme opět využívat fotografie, které mají rozměry 54×64 pixelů. Za účelem srovnání těchto dvou druhů neuronových sítí *GHA* a *APEX* jsme provedli pro každý druh neuronové sítě celkem 4 měření. Abychom zjistili, jak změna parametru počtu epoch daného druhu neuronové sítě ovlivní úspěšnost klasifikace a celkovou dobu trénování sítě, bylo nutné zafixovat dva zbývající parametry (rychlost učení a počet hlavních komponent). Uvedené dva druhy neuronových sítí byly natrénovány nad stejnou trénovací množinou. Testování těchto neuronových sítí proběhlo nad dvěma testovacími množinami (TestSet1 a TestSet2). Je důležité si uvědomit, že uvedené výsledky našich měření byly ovlivněny pevně nastavenými parametry. Je zřejmé, že změna těchto pevně nastavených parametrů ovlivní celkově chování neuronové sítě.

Výsledky klasifikace, ovlivněné změnou parametru počtu epoch, jsou shrnuty pro neuronovou síť *GHA* v tabulce č. 5. a pro neuronovou síť *APEX* v tabulce č. 6. Z tabulky č. 5. vyplývá, že se pro zafixované parametry (rychlost učení a počet hlavních komponent) s rostoucím počtem epoch zvyšuje celkový čas na trénování neuronové sítě *GHA*. Z prvního řádku této tabulky s nastaveným parametrem počet epoch sítě na hodnotu 1 je také vidět, že úspěšnost klasifikace pro testovací množinu TestSet1 byla 78%. Pro množinu TestSet2 byla 71%. Celkový čas pro trénování neuronové sítě zde byl 2 sekundy 59 milisekund. Oproti tomu ze čtvrtého řádku této tabulky plyne, že pro stejné nastavení zafixovaných parametrů (rychlost učení a počet hlavních komponent) neuronové sítě a při nastavení počtu epoch 100, se celková doba trénování zvyšuje na 5 minut. Z tohoto řádku je patrné, že i když hodnota parametru počtu epoch byla skokově zvětšena z hodnoty 1 na hodnotu 100, procentní úspěšnost pro množinu TestSet1 se příliš nezvětšila. U množiny TestSet2 se za stejných podmínek procentní úspěšnost změnila o 14%.

Z tabulky č. 6. pro neuronovou síť *APEX* je zřejmé, že pro stejnou konfiguraci sítě, jako měla neuronová síť *GHA*, došlo k poklesu celkové doby trénování sítě

a navíc došlo k růstu procentní úspěšnosti sítě.

Síť GHA	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.0008-8-1	00:02.59	78% (399)	00:04.23	71% (184)	00:02.19
54 × 64 0.0008-8-10	00:30.02	80% (409)	00:05.16	79% (205)	00:02.14
54 × 64 0.0008-8-50	02:19.37	80% (409)	00:04.56	80% (207)	00:02.17
54 × 64 0.0008-8-100	05:00.59	81% (411)	00:05.01	85% (221)	00:02.34

Tabulka 5.: Výsledky rozpoznávání pro síť GHA s variabilním parametrem počet epoch.

Síť APEX	Celkový čas trénování	TestSet1 (506)	Čas rozp. TestSet1	TestSet2 (268)	Čas rozp. TestSet2
54 × 64 0.0008-8-1	00:01.54	76% (389)	00:04.26	72% (188)	00:02.09
54 × 64 0.0008-8-10	00:18.23	83% (421)	00:05.12	76% (198)	00:02.19
54 × 64 0.0008-8-50	02:20.02	86% (436)	00:04.46	75% (195)	00:02.06
54 × 64 0.0008-8-100	03:20.09	90% (457)	00:04.55	79% (204)	00:02.26

Tabulka 6.: Výsledky rozpoznávání pro síť APEX s variabilním parametrem počet epoch.

Závěr

Hlavním smyslem této diplomové práce bylo seznámit se blíže s problematikou neuronových sítí a jejich využitím v procesu rozpoznávání objektů v obraze, který v tomto případě tvoří ucelená množina portrétních fotografií konkrétních osob. Práce obsahuje výsledky analýz a experimentů s dvěma typy neuronových sítí použitých ve fázi redukce dat. V této práci byly použity neuronové sítě *GHA* a *APEX*, které realizují analýzu hlavních komponent (PCA). Z provedených experimentů a měření za daných podmínek je vidět, že neuronová síť *APEX* se jeví v procesu redukce dimenzionality dat jako rychlejší ve srovnání se sítí *GHA* v procesu trénování. Výsledky testování použitých neuronových sítí jsou přehledně uspořádány v tabulkách, doplněných i jejich stručným zhodnocením. Z hlediska praktického využití by bylo vhodné provést tento proces rozpoznávání na daleko širším výběru portrétních fotografií (tisíce a více) a to jak ve fázi trénování tak i v procesu rozpoznávání. To by však vyžadovalo daleko výkonnější hardware. Tato práce byla realizována v domácích podmínkách na běžném osobním počítači, kde proces trénování neuronových sítí byl pro přiměřené hodnoty nastavených parametrů časově zvládnutelný. Výstupem této diplomové práce je zároveň i aplikace, která by našla svoje uplatnění v identifikaci osob v různých oblastech (policie, vojsko, letiště atd.).

Conclusions

The main purpose of this master thesis was to get acquainted with the problems of neural networks and their use in the process of recognizing objects in the image, which in this case consists of a compact set of portraits of specific individuals. The work contains the results of analyzes and experiments with two types of neural networks used in data reduction phase. In this work were used neural networks *GHA* and *APEX*, which realize Principal Component Analysis (PCA). The performed experiments and measurements under these conditions is seen that the neural network *APEX* appears in the dimensionality reduction process data faster as compared with the network *GHA* training process. Results of testing the neural networks are arranged in tables, supplemented their brief evaluation. In view of of practical use would be appropriate to carry out this process of recognizing a much wider selection of portraits (thousands and more) both in the training phase and in the recognition process. However, this would require a much more powerful hardware. This work was realized at home on a personal computer, where the process of neural network training was reasonable values for the set parameters time manageable. The output of of this master thesis is also an application that would find its application in the identification of persons in various areas (police, military, airports etc.).

Reference

- [1] Devore, J. L. *Probability and Statistics for Engineering and the Sciences*. Duxbury Press, 7.vydání 2008, 768p., ISBN - 13: 9780538733526.
- [2] Egmont-Petersen M., Ridder D., Handels H. *Image processing with neural networks—a review*. Pattern Recognition 35, 2002, pp. 2279–2301
- [3] Fauseett, L. *Fundamentals of neural networks architectures, algorithms and applications*. Pearson, December 19, 1993, 461p., ISBN-13: 978-0133341867
- [4] Haykin, S. *Neural network A Comprehensive Foundation Second Edition*. Pearson Education, India, 1999.
- [5] Hen H., Hwang J. *Handbook of neural network signal processing*. CRC Press, New York, 2001, 384p., ISBN: 0-8493-2359-2.
- [6] Kung S.Y, Diamaturas K. I. *A Neural network learning algorithm for adaptive principal component extraction (APEX)*. Department of Electrical Engineering, Princeton University, Princeton NJ 08544
- [7] Kung S.Y, Diamaturas K. I., Taur J. *Adaptive Principal Component Extraction (APEX) and Application*. IEEE Transaction on signal processing, Vol. 42. No. 5, May 1994
- [8] Meloun M., Militký J. *Statistická analýza experimentálních dat*. 2. vydání Praha: Academia, 2004, 928p., ISBN: 80-200-1254-0
- [9] Oravec M. *Metódy strojového učenia na extrakciu príznačkov a rozpoznávanie vzorov*. STU, Bratislava, 2012, 150p., ISBN: 978-80-227-3691-6
- [10] Rojas R. *Neural Networks: A Systematic introduction*. Springer-Verlag, Berlin, 1996.
- [11] Szeileski, R. *Computer Vision: Algorithms and Applications*. New York : Springer, 2011, 811p., ISBN: 978-1-84882-934-3.
- [12] Šíma J., Neruda R. *Teoretické otázky neuronových sítí*. MATFYZPRESS, Prague, 1996, 390p.
- [13] Turk M., Pentland A. *Eigenfaces for Recognition*. Massachusetts Institute of Technology, 1991.
- [14] Volná E. *Neuronové sítě 1*. 2. vydání. Ostrava, 2008, 86p.
- [15] Žára J., Beneš B., Sochor J., Felkel P. *Moderní počítačová grafika*. 2. vydání. Brno: Computer Press, 2004, 576p., ISBN: 80-251-0454-0.
- [16] <http://cmp.felk.cvut.cz/spacelib/faces/>

6. Obsah přiloženého CD

V samotném závěru práce je uveden stručný popis obsahu přiloženého CD, tj. závazné adresářové struktury, důležitých souborů apod.

`bin/`

Instalátor programu `IMAGEPATTERNRECOGNITION` a další programy spustitelné přímo z CD. Adresář obsahuje i všechny potřebné knihovny a další soubory pro bezproblémové spuštění programu.

`doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PřF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`src/`

Kompletní zdrojové texty programu `IMAGEPATTERNRECOGNITION` se všemi potřebnými (převzatými) zdrojovými texty, knihovnami a dalšími soubory pro bezproblémové vytvoření spustitelných verzí programu.

`readme.txt`

Instrukce pro instalaci a spuštění programu `IMAGEPATTERNRECOGNITION`, včetně požadavků pro jeho provoz.

Navíc CD obsahuje:

`data/`

Ukázková a testovací data použitá v práci a pro potřeby obhajoby práce.

`install/`

Instalátory aplikací, knihoven a jiných souborů nutných pro provoz programu.

`literature/`

Některé položky literatury odkazované z dokumentace práce.

U veškerých odjinud převzatých materiálů obsažených na CD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.