

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

VYHODNOCENÍ ÚSPĚŠNOSTI DETEKCE OBJEKTŮ  
V OBRAZE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

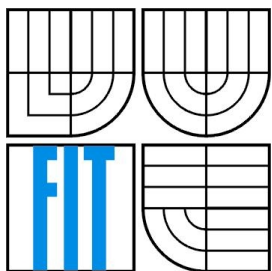
AUTOR PRÁCE  
AUTHOR

BEDŘICH ČERNOŠEK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# VYHODNOCENÍ ÚSPĚŠNOSTI DETEKCE OBJEKTŮ V OBRAZE

EVALUATION OF OBJECT DETECTION IN IMAGE

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

BEDŘICH ČERNOŠEK

VEDOUCÍ PRÁCE  
SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2015

## Abstrakt

Cílem této bakalářské práce bylo navrhnout způsob vyhodnocení úspěšnosti detekce objektů v obraze. Výsledkem této práce bylo vytvořit program, který na vhodném příkladu provede vyhodnocení úspěšnosti detekce objektů a intuitivně zobrazí výsledek uživateli. Úkolem bylo navrhnout vhodné experimenty a datový set pro ověření správnosti vyhodnocení. Součástí práce bylo nalézt optimální parametry detektoru pro detekci obličejů a optimální předzpracování fotografie před detekcí obličejů.

## Abstract

The main goal of this bachelor's thesis was to propose the evaluation method of object detection. Result of this work was to create a program which performs the evaluation of object detection on suitable data sample and intuitively displays result to user. The task was to propose suitable experiments and dataset for proving correctness of evaluation. Part of this work was to find optimal parameters for face detection and optimal photo preprocessing before the face detection.

## Klíčová slova

detekce objektů v obraze, detekce obličejů, úspěšnost detekce, parametry detektoru, předzpracování obrazu, AdaBoost, WaldBoost, klasifikace, trénování, vzorkování, převzorkování, interpolace

## Keywords

object detection in image, face detection, detection success, detector parameters, image preprocessing, AdaBoost, WaldBoost, classification, training, sampling, resampling, interpolation

## Citace

Bedřich Černošek: Vyhodnocení úspěšnosti detekce objektů v obraze, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Vyhodnocení úspěšnosti detekce objektů v obraze

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana prof. Dr. Ing. Pavla Zemčíka.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Bedřich Černošek

18.5.2015

## Poděkování

Chci poděkovat vedoucímu této práce panu prof. Dr. Ing. Pavlu Zemčíkovi za poskytnutí užitečných rad a pomoci při tvorbě této práce. Dále bych chtěl poděkovat své rodině za podporu ve studiu a při tvorbě této práce.

© Bedřich Černošek, 2015

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	1
1 Úvod.....	2
2 Shrnutí dosavadního stavu.....	4
2.1 Detekce.....	4
2.2 AdaBoost.....	4
2.3 WaldBoost.....	5
2.4 Konvoluční neuronové sítě.....	7
2.5 Rozpoznávání.....	8
3 Zhodnocení současného stavu a plán práce.....	9
3.1 Zhodnocení současného stavu.....	9
3.2 Zadání práce.....	9
3.3 Návrh vyhodnocení úspěšnosti detekce.....	9
3.4 Návrh experimentů.....	11
3.5 Požadavky na řešení.....	11
4 Popis práce a zhodnocení výsledků.....	13
4.1 Popis experimentů.....	13
4.2 Výsledky experimentů.....	19
4.3 Vyhodnocení úspěšnosti detekce.....	27
4.4 Zobrazení výsledků experimentů.....	29
4.5 Tvorba datasetu.....	30
4.6 Zhodnocení výsledků.....	30
5 Závěr.....	32
Literatura.....	33
Seznam obrázků.....	35
Seznam příloh.....	36

# 1 Úvod

Detekce objektů v obraze je v současné době velmi diskutovaná disciplína zpracování obrazu. Zpracování obrazu se zabývá zpracováním obrazových dat (fotografie, video), která jsou pořízena pomocí speciálního zařízení (kamera, fotoaparát). V rámci zpracování lze provádět vylepšení obrazu, úpravu obrazu nebo získávat informace z obrazu. Detekce objektů v obraze je technologie, která je pozitivně přijímána spíše technickými nadšenci a bezpečnostními experty. Tato moderní technologie má však i své odpůrce a to například z důvodu narušování soukromí nebo krádeže identity. Jak již název přímočaře napovídá, jedná se o metodu, postup, algoritmus, který prochází fotografií nebo videozáznam, na kterých vyhledává objekty. Tyto objekty označuje, respektive získává souřadnice, výšku a šířku objektu nacházejícího se na fotografii. Jaké objekty detektor detekuje závisí na trénování detektoru. Detektor může být například natrénován na detekci obličejů, postav, tvarů, předmětů a podobně. Detekce objektů se skládá z několika kroků, které jsou konkrétně popsány v následující kapitole. Tyto kroky jsou závislé na konkrétním použitém algoritmu detekce nebo přístupu. Stěžejní částí detekce, která je součástí každého detektoru je klasifikace, na základě které se rozhoduje o výsledné detekci. Detekce objektů v obraze přímo souvisí s rozpoznáváním objektů, které následuje po detekci objektů. Díky rozpoznávání objektů je možné rozeznat konkrétního člověka, předmět nebo tvar z detekovaného objektu v obraze. Pro detekci objektů v obraze je podstatná rychlost detekce a to především pro zpracování obrazu z videozáznamu nebo kamerového záznamu v reálném čase. Rychlost detekce objektů v obraze je závislá na dvou aspektech. Nejen na rychlosti detekčního algoritmu, ale i na rychlosti zpracování použitého hardwaru. Detekci objektů lze provádět na několika hardwarových platformách. S pomocí procesoru (CPU), grafického procesoru (GPU) nebo speciálně vytvořeného obvodu vypáleného do programovatelného hradlového pole (FPGA). Vytvořením specializovaného obvodu lze dosahovat výsledků detekce nejrychleji.

Detekce objektů v obraze nachází široké uplatnění v různých odvětvích. Tato technologie je implementována v široké řadě výrobků a služeb a používá ji dnes většina technicky gramotných lidí na světě. Často se detekce objektů využívá pro detekování obličejů v obraze. S detekcí obličejů v obraze se nejčastěji člověk může setkat ve fotoaparátech nebo kamerách. Jak v samotných zařízeních, tak i ve fotoaparátech nebo kamerách, které jsou součástí mobilních telefonů, tabletů, notebooků a podobných zařízení. Součástí fotoaparátů nebo kamer již však není rozpoznávání, které v tomto případě nemá význam. Detekce obličejů v těchto zařízeních slouží k zaostření fotografie v dané oblasti, kde se nachází obličej nebo obličeje. Dalším případem, kde se člověk může často setkat s detekcí obličeje v obraze jsou sociální sítě. V této oblasti již má význam i rozpoznávání, které však není ve většině případů na sociálních sítích plně spuštěno. Cílem je na nahané nebo pořízené fotografii uživatelem detekovat obličeje a nabídnout uživateli možnost označit osoby, které zná. V budoucnosti se počítá i s automatickým rozpoznáním obličeje, pokud tomu nebude v EU bránit legislativa. V USA již je v provozu například aplikace pro rozpoznání obličejů DeepFace na sociální síti Facebook. Tato aplikace slibuje rozpoznání obličejů na úrovni rozpoznání člověkem. Detekce a rozpoznání obličeje z obrazu může také sloužit k autentizaci uživatele na určitém zařízení nebo při vstupu do místnosti, budovy nebo objektu. Může být využito samostatně, jako součást biometrické autentizace nebo v kombinaci se zadáním hesla. Velmi populární je také použití detekce obličeje a postavy člověka k ovládní programů a her s použitím speciálního hardware vybaveného kamerou a senzory. Příkladem je sensorové zařízení Kinect od společnosti Microsoft, které může být součástí herní konzole Xbox. Asi nejvíce diskutovaným použitím detekce a rozpoznávání obličeje z obrazu je při kontrolách lidí na veřejných místech. Příkladem může být hraniční pasová kontrola, letištní kontrola a podobně. Při použití detekce a rozpoznávání v tomto případě můžeme zabránit pohybu nebezpečné osoby nebo odhalit a pomoci zajistit hledanou osobu. To lze však realizovat pouze za předpokladu vybudování kvalitní databáze obličejů osob.

Detekce objektů v obraze může taky člověku pomáhat s diagnózou, léčbou nebo záchranou lidských životů. V medicíně lze detekci objektů v obraze použít při zpracování medicínských snímků. Například lze použít detekci objektů pro detekci tepny z ultrazvukového snímku nebo detekci kloubu

ze snímku magnetické rezonance a podobně. Dalším příkladem, kde může být detekce objektů v obraze prospěšná, je doprava. Detekce dopravních značek může pomoci řidiči dodržovat předepsanou rychlost. Detekce jízdního pruhu může řidiče informovat případně probudit při změně jízdního pruhu bez signalizace směrovými světly vozidla. Detekce postav může zabránit řidiči střetu s chodci, cyklisty a dalšími účastníky dopravního provozu. Detekce objektů v obraze může také usnadňovat práci statistikům v případě, že detekuje a zaznamená sledovanou událost. Takovou událostí může být průchod osoby, průjezd auta a podobně.

Moje bakalářská práce se především zaměřuje na detekci obličejů v obraze. Přínosem mojí bakalářské práce je navržení automatizovaného vyhodnocení úspěšnosti detekce objektů v obraze. Tohle řešení lze aplikovat na detekci jakýchkoliv objektů v obraze. Dalším přínosem této bakalářské práce je poskytnutí přehledných výsledků experimentů, ze kterých uživatel získá efektivní parametry pro nastavení detektoru obličejů a optimální předzpracování fotografie před detekcí. Správnost algoritmu vyhodnocení úspěšnosti detekce je ověřena na vytvořeném datasetu.

Tato bakalářská práce je rozdělena na pět částí respektive kapitol. První krátkou kapitolou je úvod bakalářské práce, ve které se čtenář právě nachází. Seznamuje jej s problematikou detekce objektů v obraze obecně a uvádí jej do problematiky této bakalářské práce. Druhá kapitola je teoretická a zabývá se vysvětlením dostupných a použitých algoritmů, postupů a metod v oblasti detekce objektů v obraze. Tato část poskytuje čtenáři nezbytný teoretický základ do dané problematiky, který je nutný pro pochopení návrhu a řešení zadaného problému. Konkrétně se zabývá klasifikací a trénováním, což jsou základní kroky při detekci objektů v obraze a také rozpoznáváním, které na detekci navazuje. Rozebrány jsou používané přístupy a algoritmy k detekci objektů. Třetí kapitola je zaměřena na návrh řešení zadaného problému. Nejprve je v této části provedeno shrnutí současného stavu a je podrobně rozebráno zadání bakalářské práce. Dále se zabývá návrhem vyhodnocení úspěšnosti detekce objektů v obraze. Také se zabývá návrhem experimentů s parametry detektoru a předzpracováním fotografie před detekcí objektů. Na závěr této části jsou uvedeny požadavky na použitý software, který je potřebný pro řešení práce. Čtvrtá kapitola je praktická a zaměřuje se na způsob dosažení cíle a splnění této bakalářské práce. V této části je popsán postup práce a dosažené výsledky. Je zde představen a konkrétně popsán způsob vyhodnocení úspěšnosti detekce objektů v obraze. Dále jsou konkrétně popsány jednotlivé experimenty s nastavením detektoru a předzpracováním fotografie před detekcí. Popsána je zde i tvorba datasetu, na kterém byla činnost algoritmu ověřena. Na závěr kapitoly jsou zhodnoceny dosažené výsledky. Poslední pátou kapitolou mé bakalářské práce tvoří závěr, kde je popsán cíl práce a skutečnost, zda se jej podařilo dosáhnout. Je zde krátce shrnuto vytvořené řešení, které bylo konkrétně popsáno v předchozí čtvrté kapitole. Uvedeny jsou také výsledky provedených experimentů. Tato kapitola se také zabývá dalším možným vylepšením a pokračováním mé bakalářské práce.

## 2 Shrnutí dosavadního stavu

V této části práce jsou konkrétně popsány algoritmy, které se používají pro implementaci detekce objektů v obraze. Především je věnována pozornost algoritmům AdaBoost, WaldBoost a konvolučním neuronovým sítím. Jsou popsány jednotlivé kroky, které se provádějí v rámci výše zmíněných detekčních algoritmů. Součástí popisu metod jsou také různá vylepšení, která zvyšují efektivitu činnosti nebo kvalitu výsledku detekce. Na závěr kapitoly je popsáno rozpoznávání detekovaného objektu. Tato kapitola slouží pro teoretický náhled do problematiky detekce a její osvojení je nezbytné pro porozumění praktické části mé bakalářské práce.

### 2.1 Detekce

Detekce objektů v obraze se skládá převážně z procesu trénování neboli učení a klasifikace. Na detekci objektů v obraze navazuje rozpoznávání objektů. Při řešení detekce objektů v obraze lze použít více přístupů. Například lze použít metody založené na strojovém učení nebo biologické algoritmy a podobně. Mezi algoritmy založené na strojovém učení patří algoritmus AdaBoost a WaldBoost, které jsou níže podrobně popsány. Dále je také popsán algoritmus konvolučních neuronových sítí, které patří mezi biologické algoritmy.

### 2.2 AdaBoost

Boosting (posílení) je přístup založený na strojovém učení. Hlavní myšlenkou je vytvořit vysoce přesné predikční pravidlo, které kombinuje mnoho relativně slabých a nepřesných pravidel [2]. Algoritmus AdaBoost vytvořený Yoavem Freundem a Robertem Schapirem byl první boosting algoritmus, který se stal jedním s nejpoužívanějších, nejstudovanějších a nejdiskutovanějších [2]. Existuje mnoho variant tohoto algoritmu. Lze jej využít v kombinaci s jiným algoritmem učení a tím zvýšit jeho účinnost. Výstup učících algoritmů (tzv. slabých učících se) je kombinován do váženého součtu, který reprezentuje finální výstup posíleného klasifikátoru [2]. AdaBoost je zkrácený výraz pro Adaptive Boosting (adaptivní posílení). AdaBoost je adaptivní ve smyslu, že jednotliví slabí učící se jsou vyladěni ve prospěch jejich instancí [1]. Algoritmus se adaptuje na úroveň chyby jednotlivé slabé hypotézy [1]. Jednotliví učící se mohou být slabí, ale jsou úspěšnější než náhodné hádání [1] [2]. Úspěšnost slabého klasifikátoru by měla být větší než 50%, jako je u náhodného hádání [1] [2]. Ve skutečnosti je úspěšnost přibližně kolem 60%. Úroveň chyby je menší než 0.5. Tyto hodnoty jsou uvažovány pro binární klasifikaci.

Jak funguje algoritmus AdaBoost navržený Freundem a Schapirem je níže popsáno. Také je uveden pseudokód tohoto algoritmu. Je dáno  $m$  označených trénovacích vzorků, kde  $x_i$  jsou v nějaké doméně  $X$  a označení  $y_i$  náleží  $\{-1,1\}$ . Následně probíhá inicializace distribuce  $D_1$ .  $D_1$  je počáteční inicializovaná distribuce, kde  $m$  je počet vzorků a  $i$  je daný index vzorku. V každé iteraci cyklu  $t$  dochází k trénování slabého učícího se s použitím distribuce, která je vypočítána nad  $m$  trénovacími vzorky. Daný slabý učící se neboli algoritmus slabého učení je použit k vyhledání slabých hypotéz, kde cíl slabého učícího se je najít slabé hypotézy s nízkou váhou chyby vzhledem k distribuci  $D_t$ . Trénování silného klasifikátoru probíhá v cyklu. Jeden slabý klasifikátor je vybrán a přidán k součtu v každé iteraci cyklu. Dále se vybere váha. Aktualizuje se hodnota distribuce  $D_t$  pro  $i$ . Hodnota  $Z_t$  je normalizační faktor. Konečná nebo kombinovaná hypotéza  $H$  vypočítá znaménko vážené kombinace slabých hypotéz.  $H$  je tedy vypočítáno jako vážená většinová volba slabých hypotéz  $h_t$ , kde každá má přiřazenou váhu. Hodnota  $h_t$  je slabá hypotéza neboli slabý klasifikátor, který bere  $x$  jako vstup a vrací reálnou hodnotu výsledku indikující třídu objektu.  $\alpha$  je přiřazená váha hypotézy. Znaménko výstupu identifikuje predikovanou třídu objektu a absolutní hodnota vrací důležitost v klasifikaci. V každé iteraci trénovacího procesu je váha přiřazena ke každému vzorku z trénovací sady, který se



rovná chybě na daném vzorku. Tyhle váhy mohou být použity pro informaci při trénování slabého algoritmu učení [1] [2].

$$\begin{aligned}
 & \text{Dány vzorky : } (x_1, y_1), \dots, (x_m, y_m), \text{ kde } x_i \in \mathcal{X}, y_i \in \{-1, +1\} \\
 & \text{Inicializuj : } D_1(i) = 1/m \text{ pro } i = 1, \dots, m \\
 & \text{pro } (t = 1, \dots, T): \\
 & \quad - \text{trénuj slabého učícího se s použitím distribuce } D_t \\
 & \quad - \text{získej slabou hypotézu } h_t : \mathcal{X} \rightarrow \{-1, +1\} \\
 & \quad - \text{vyber } h_t \text{ s nízkou váženou chybou } \varepsilon_t = \Pr_i \sim D_t [h_t(x_i) \neq y_i] \\
 & \quad - \text{vyber váhu } \alpha_t = 1/2 \ln(1 - \varepsilon_t / \varepsilon_t) \\
 & \quad - \text{aktualizuj, pro } i = 1, \dots, m : D_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i h_t(x_i)) / Z_t \\
 & F(x) = \sum_{t=1}^T \alpha_t h_t(x) \\
 & F(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)
 \end{aligned}$$

## Haarovy příznaky

Používají se jako jednoduchý klasifikátor. Pro klasifikaci se využívají prahové hodnoty. Haarovy příznaky jsou vygenerovány odvozením z obdélníků. Tyto obdélníky jsou rozděleny podle typu. Mohou být hranové příznaky, čárové příznaky nebo středové příznaky. Bílá oblast obdélníku má nastavenou váhu -1. Váha černé oblasti se vypočítá jako podíl bílé oblasti a černé oblasti. Hodnota příznaku je rovna odezvě. Haarovy příznaky se používají v kombinaci s integrálním obrazem a kaskádou klasifikátoru u AdaBoost detektoru navrženého Violou a Jonesem [3].

## Integrální obraz

Vylepšením algoritmu je tzv. integrální obraz. Tohle vylepšení algoritmu AdaBoost pochází od Violy a Jonese. Jedná se o novou reprezentaci obrazu. Tato reprezentace slouží k rychlejšímu výpočtu respektive vyhodnocení. Integrální obraz se používá u Haarových příznaků. Obraz je převeden na integrální místo výpočtu součtu hodnot pixelů v definovaných obdélnících. Je dosaženo rychlejšího výpočtu odezvy Haarova příznaku na vstupní obraz [3].

## Kaskáda klasifikátorů

Jedná se o vylepšení algoritmu AdaBoost taktéž od Violy a Jonese. Umožňuje kombinaci klasifikátorů. Umožňuje velmi rychle odstranit pozadí obrázku, kde se nenachází obličej. Zaměřuje se na oblasti, kde by se mohly obličej vyskytovat. Skládá se z tzv. stupňů, které jsou složeny ze slabých klasifikátorů. Každý stupeň je ve své podstatě silný klasifikátor, který má prahovou hodnotu. Podle této prahové hodnoty se rozhoduje, jestli je daný vzorek pozitivní nebo negativní. Stupeň kaskády vyhodnotí vzorky a pozitivní odešle na další stupeň. Každý následující stupeň obsahuje slabé klasifikátory předcházejícího. Z hlediska efektivity klasifikace je vhodné, aby první stupeň vyřadil co nejvíce vzorků [3].

## 2.3 WaldBoost

WaldBoost algoritmus integruje algoritmus AdaBoost, respektive výběr a klasifikaci tohoto algoritmu [5] [6]. Vylepšením oproti AdaBoostu je tzv. rozhodovací strategie SPRT. Nazývána je také

jako Waldovo sekvenční rozhodování nebo přesněji sekvenční test poměru pravděpodobnosti [5] [6]. Algoritmus WaldBoost je kompromisem mezi efektivním časem detekce a úrovní chyby [6].

Klasifikace algoritmu WaldBoost probíhá tak, že jsou dány prahy tzv. thresholdy, slabý klasifikátor nebo klasifikátory, uživatelem zvolená hodnota v případě neúspěšné klasifikace a iterace cyklu  $t$ . Vstupem je objekt  $x$ , který má být klasifikován. Po daný počet iterací cyklu  $t$  běží program. Nejprve je provedeno rozhodnutí sekvenčního testu. Pokud slabý klasifikátor  $H_t$  překročí nastavený threshold, je učiněno rozhodnutí. To znamená, že je vybrán další slabý klasifikátor a přidán k součtu. Pokud není rozhodnutí učiněno v daných iteracích cyklu, dochází k nastavení hodnoty určené uživatelem. Z praktických důvodů je použito pouze omezené množství klasifikátorů. Rozhodnutí o výběru klasifikátoru provádí sekvenční rozhodovací strategie, respektive procedura sekvenčního testu [6].

*jsou dány :  $h_t, \Theta_A^t, \Theta_B^t, \gamma, (t=1, \dots, T)$   
 vstup : objekt  $x$ , který má být klasifikován  
 pro :  $t=1, \dots, T$  (vykonání SPRT rozhodnutí)  
 – pokud :  $H_t(x) \geq \Theta_B^t$ , klasifikuj  $x$  do třídy +1 a ukonči  
 – pokud :  $H_t(x) \leq \Theta_A^t$ , klasifikuj  $x$  do třídy -1 a ukonči  
 konec cyklu  
 pokud :  $H_T(x) > \gamma$ , klasifikuj  $x$  do třídy +1, jinak do třídy -1*

Učení algoritmu WaldBoost s využitím Bootstrappingu probíhá tak, že je k dispozici označená trénovací sada vzorků, falešně pozitivní hodnota  $\beta$  a falešně negativní hodnota  $\alpha$  výstupního klasifikátoru. Tyto dvě hodnoty jsou použity pro výpočet thresholdů A a B. Dále probíhá inicializace vah a nastavení prahů A a B. Trénování probíhá v cyklu, kde první krok je standardní AdaBoost vyhledávání pro nejlepší slabý klasifikátor. Dále jsou nalezeny poměr pravděpodobnosti a thresholdy  $\Theta_A$  a  $\Theta_B$ . Na základě thresholdů je trénovací set ořezán. Poté je trénovací set zvětšen novými snímky. Výstupem je silný klasifikátor a prahy [6].

*vstup :  $(x_1, y_1), \dots, (x_l, y_l); x_i \in \mathcal{X}, y_i \in \{-1, +1\}$ ,  
 požadované konečné false negative  $\alpha$  a false positive  $\beta$  úrovně  
 inicializace :  $w_1(x_i, y_i) = 1/l$   
 nastavení :  $A = (1 - \beta)l$  a  $B = \beta/(1 - \alpha)$   
 pro :  $t=1, \dots, T$   
 – vyber slabý klasifikátor  $h_t$   
 – odhad poměru pravděpodobnosti  $R_t$   
 – nalezení prahů  $\Theta_A^t$  a  $\Theta_B^t$   
 – zahození vzorků z trénovací sady, které  $H_t \geq \Theta_B^t$  nebo  $H_t \leq \Theta_A^t$   
 – získat nová data do trénovací sady  
 konec cyklu  
 výstup : silný klasifikátor  $H_T$  a prahy  $\Theta_A^t$  a  $\Theta_B^t$*

## Sekvenční rozhodovací strategie SPRT

Sekvenční rozhodovací strategie  $S^*$  funguje následovně. Provádí se jedno měření v čase a kroku  $t$ . Vytváří na základě měření rozhodnutí  $S_t$ . Rozhodnutí je charakterizováno chybou  $\alpha$  a  $\beta$ . Hodnota  $\alpha$  znamená falešně negativní chybu a hodnota  $\beta$  falešně pozitivní chybu. Strategie je charakterizována úrovní chyby a průměrným výpočetním časem. U SPRT se využívá silného klasifikátoru se sekvencí prahů. Hodnota falešně pozitivní je nastavena na nulu. Falešně negativní je

malá konstanta, pokud možno co nejmenší. Pokud je  $R_t$  menší nebo rovno nule výsledek klasifikace je +1. Pokud je  $R_t$  větší nebo rovno  $1/\alpha$  výsledek klasifikace je -1. Pokud je  $R_t$  jiné, použije se další klasifikátor [6].

## LBP příznaky

LBP (Local Binary Pattern) je dalším vylepšením, které se u WaldBoost algoritmu používá. Jsou to tzv. lokální binární vzory [4] [5]. Zpracování LBP příznaků probíhá následovně. Nejprve se obrázek převede do odstínů šedi tzv. grayscale. Poté se postupně prochází všechny pixely obrazu kromě krajních pixelů. Provede se prahování tzv. thresholding. Daný pixel se vynásobí vahou. Získáváme LBP hodnotu pro centrální bod. Ze všech těchto hodnot je vytvořen histogram četnosti výskytu tzv. příznak obrázku. LBP příznaky jsou jednodušší a rychlejší než Haarovy příznaky [4]. Tato implementace je vhodná pro použití v FPGA [4]. Tudíž ji lze využít pro detekci v reálném čase [4]. Dochází při ní k dedukci slabých klasifikátorů. LBP je neparametrický operátor, později strukturovaný operátor, který zajišťuje obrazovou analýzu [4]. Pracuje se s maticí 9 hodnot pixelů obrázku [4]. Tato matice zahrnuje centrální pixel a 8 okolních pixelů (oblast 3x3 pixely). Výsledkem je set konvolucí jejíž vstupem jsou pixely v šedé škále. Výhodnocení LBP operátoru probíhá porovnáním každého okrajového pixelu s centrálním [4] [5]. Pokud je menší, výsledkem je logická 0, jinak logická 1. Výsledkem je 8x1bit=1byte [5]. Může tedy být 256 různých výsledků. Používá se také MB\_LBP, tzv. multi-block LBP. Používá větší oblast a tudíž je i přesnější, preciznější. Velikost detekčního okna (oblasti, okolí) je většinou 24x24 pixelů pro detekci obličejů v obraze, jinak většinou 6x6 pixelů [4].

## 2.4 Konvoluční neuronové sítě

Neuronové sítě jsou označovány jako vícevrstvý perceptron. Tyto sítě jsou považovány za jeden z nejjednodušších klasifikátorů [8]. Perceptron je nejjednodušším modelem neuronové sítě, který se skládá z jednoho neuronu. Lze jej použít na množiny, které jsou lineárně separovatelné. Rozšířením je vícevrstvý perceptron, který se skládá z více vrstev výpočetních jednotek. Každý neuron v jedné vrstvě je přímo spojen s neuronem na další vrstvě. Rozšířením těchto sítí jsou konvoluční neuronové sítě [8].

Koncept konvolučních neuronových sítí představili roku 1995 Yann LeCun a Yoshua Bengio [8]. Konvoluční neuronové sítě jsou hladové sítě se schopností extrahovat topologické vlastnosti ze vstupního obrázku bez jakéhokoliv předzpracování [8] [9]. Tyhle sítě integrují výběr příznaků do trénovacího procesu [8] [10]. Umí rozpoznat vzory s extrémní variabilitou, velkým zkreslením, rotací, transformací, zvětšením, zmenšením, stlačením výšky, stlačením šířky, šumem a podobně [8] [10]. Konvoluční neuronové sítě kombinují tři myšlenky pro potlačení vlivu posunu, zvětšení a zkreslení. Mezi tyto tři stavební prvky patří lokální citlivá pole, sdílené váhy a prostorové nebo dočasné sub vzorkování [8] [9]. Síť je obvykle trénovaná jako standardní neuronová síť zpětnou propagací [8] [9] [10]. Vrstvy sítě se střídají mezi vrstvami konvoluce s mapou příznaků  $C_{k,l}$  a nepřekrývajícími sub vzorkovacími vrstvami s mapou příznaků  $S_{k,l}$  [8] [10].

$$C_{k,l}^i = g(I_{k,l}^i * W_{k,l} + B_{k,l})$$

$$S_{k,l}^i = g(I_{k,l}^i \downarrow w_{k,l} + Eb_{k,l})$$

Kde  $g(x)$  je rovna  $\tanh(x)$  a je aktivační sigmoidní funkcí,  $B$  a  $b$  jsou tzv. biasy,  $W$  a  $w$  jsou váhy,  $I_{k,l}$  je  $i$ -tý vstup a  $E$  je matice elementů. Hvězdička znázorňuje operaci dvoudimenzionální konvoluce. Šipka dolů znázorňuje operaci převzorkování. Velká písmena značí matice, malá písmena značí skaláry. Konvoluční vrstva extrahuje příznaky z citlivých polí na předcházející vrstvě [8]. Konvoluční vrstva je složena z roviny neuronů, které jsou označovány jako mapa příznaků, která detekuje

specifický příznak [8]. V síti s konvolučním jádrem velikosti 5x5 má každá jednotka 25 vstupů napojených na oblast 5x5 předchozí vrstvy, která je nazývána lokální citlivé pole [8]. Trénovaná váha je přiřazena ke každému spojení, ale všechny jednotky mapy příznaků sdílí stejné váhy [8]. Tento příznak, který povoluje redukci počtu trénovatelných parametrů se nazývá technika sdílené váhy a je aplikována na všech vrstvách konvoluční neuronové sítě [8] [9].

## 2.5 Rozpoznávání

Rozpoznávání obličejů může být využíváno k autentizaci a identifikaci detekovaného obličeje. První algoritmy pro rozpoznávání obličejů používaly jednoduché geometrické modely [11]. V dnešní době se proces rozpoznání obličeje vyvinul v sofistikovanou matematickou reprezentaci a porovnávání [11]. Rozpoznávání je relativně nový koncept. První poloautomatizovaný systém rozpoznávání byl vytvořen v sedmdesátých letech minulého století. Vyžadoval účast administrátora, který určil příznaky obličeje na fotografii [11]. Mezi tyto příznaky patří oči, uši, nos a ústa. Z těchto informací se poté vypočítaly vzdálenosti a poměry, které se porovnávaly s referenčními daty [11]. V osmdesátých letech minulého století byly používány specifické subjektivní značkovače [11]. Tyto značkovače již zahrnovaly barvu vlasů, tloušťku rtů a podobně. Pořád však bylo nutné tyto informace určovat manuálně. V roce 1988 byl poprvé použit k rozpoznávání princip dílčí analýzy a lineární algebra, což bylo považováno za milník v rozpoznávání [11]. V roce 1991 byla poprvé použita technika Eigenfaces (vlastní tváře), která je pojmenovaná po množině vlastních vektorů [11] [12] [13]. Vlastní vektor dané transformace označuje nenulový vektor, jehož směr se při transformaci nemění. Změna velikosti vektoru se nazývá vlastní hodnota. Množina vlastních vektorů se nazývá vlastní prostor transformace. Vlastní vektory jsou odvozeny z kovarianční matice rozdělení pravděpodobnosti nad vysokodimenzionálním vektorovým prostorem fotografie obličeje [12] [13].

V dnešní době se využívají dva přístupy k rozpoznávání detekovaných obličejů. První z nich je geometrický přístup, který je založen na příznacích (feature) [11]. Druhý přístup je fotometrický a je založen na pohledu (view) [11]. Existuje velká spousta algoritmů k rozpoznávání detekovaných obličejů. V této kapitole jsou popsány rozšířené a hojně studované algoritmy. Mezi tyto algoritmy patří PCA, LDA a EBGGM.

PCA (Principal Component Analysis) je metoda, která používá techniku Eigenfaces. Je nutné, aby rozpoznávaná fotografie a fotografie datasetu byly stejné velikosti. Fotografie musí být proto nejprve normalizovány. Normalizace se využívá pro zvýraznění linie očí a úst obličeje. Poté se provádí redukce dimenze dat. V podstatě probíhá komprese, která vrací efektivní nízkodimenzionální strukturu vzoru obličeje. V této redukci se odstraňují informace, které nejsou pro rozpoznávání užitečné nebo použitelné. Výsledkem je precizní dekompozice struktury obličeje na ortogonální části zvané Eigenfaces. Každá fotografie obličeje je reprezentována jako vážený součet (vektor příznaků) vlastních obličejů, který je uložen jako jednodimenzionální pole. U rozpoznávané fotografie a fotografie z datasetu se porovnávají vzdálenosti mezi jejich vektory příznaků [11].

LDA (Linear Discriminant Analysis) je statistický přístup pro klasifikaci vzorků neznámých tříd založený na trénování vzorků se známými třídami. Tahle technika se zaměřuje na maximalizaci odchylky mezi třídami a minimalizaci odchylky uvnitř třídy. Problém nastává v případě vysokodimenzionálních dat, která v případě malého množství trénovacích vzorků nelze porovnat [11].

EBGGM (Elastic Bunch Graph Matching) je přístup založený na konceptu, na jehož základě mají fotografie mnoho nelineárních charakteristik, které předchází přístupy neuvažovaly. Jako například osvětlení, pozice objektu, výraz obličeje a podobně. Gaborova vlnková transformace vytváří architekturu dynamických spojení, která zobrazují tvář na elastické mřížce (síti). Gaborův uzel vyznačuje chování daného pixelu. Je to výsledek konvoluce obrazu pomocí Gaborova filtru, který detekuje tvary a extrahuje příznaky pro zpracování obrazu. Rozpoznání je závislé na podobnosti výstupu Gaborova filtru v každém Gaborově uzlu [11].

## 3 Zhodnocení současného stavu a plán práce

Tato kapitola nejprve hodnotí aktuální stav práce, ve kterém se nachází. Následně je podrobně rozebráno zadání bakalářské práce, kde se zaměřuje na vymezené cíle. Další podkapitoly se již zabývají způsobem dosažení cílů. Pro každý bod zadání bakalářské práce byl vybrán a navržen vhodný postup řešení. Především byl navržen způsob vyhodnocení úspěšnosti detekce objektů v obraze. Byly navrženy experimenty a dataset pro ověření správnosti algoritmu vyhodnocení úspěšnosti detekce objektů. Pro řešení práce jsou zvoleny vhodné programy a nástroje. Vybrán je také programovací jazyk pro implementaci algoritmu.

### 3.1 Zhodnocení současného stavu

V současném stavu, ve kterém se práce nachází, byly provedeny následující kroky. Byly popsány přístupy, kterými lze řešit detekci objektů v obraze. Především byl popsán přístup založený na strojovém učení a přístup založený na biologických algoritmech. Konkrétně byly rozepsány používané algoritmy pro implementaci detektoru objektů v obraze. Z algoritmů založených na strojovém učení byly konkrétně vysvětleny algoritmy AdaBoost a WaldBoost. Z algoritmů založených na biologických algoritmech byly popsány konvoluční neuronové sítě. Pozornost byla také věnována rozpoznávání objektů, které následuje po detekci. Rozpoznávání detekovaných objektů není součástí zadání bakalářské práce, jeho popis slouží pouze pro doplnění informací. Je k dispozici zadání bakalářské práce, které je v následující podkapitole podrobně rozebráno. V této situaci bylo nutné navrhnout vhodné postupy a metody pro dosažení cíle a splnění bakalářské práce. Bylo potřeba vybrat programy a nástroje pro tvorbu práce. Důležité bylo také zvolit vhodný programovací jazyk pro implementaci algoritmů. Výběr těchto prostředků byl popsán na závěr této kapitoly. Před zahájením práce bylo nutné navrhnout způsob vyhodnocení úspěšnosti detekce a experimenty. Poté již bylo možné aktivně řešit zadání práce. Řešení práce je popsáno v následující kapitole.

### 3.2 Zadání práce

Prvním bodem zadání je prostudovat dostupnou literaturu a software na téma detekce objektů v obraze. Požadováno je zaměřit se především na metody respektive algoritmy typu AdaBoost a WaldBoost, alternativně na konvoluční neuronové sítě. Druhým bodem zadání je vybrat vhodný software pro detekci obličejů v obraze, nejlépe pro metodu WaldBoost, která bude implementována na platformě PC. Předpokladem je, že použitý software detektoru bude dobře parametrizovatelný. To znamená, že bude podporovat možnost širšího nastavení parametrů detektoru pro zjištění optimálních hodnot těchto parametrů detekce. Třetím bodem zadání je navrhnout způsob vyhodnocení úspěšnosti detekce objektů v obraze a popsat jeho vlastnosti. Tento bod zadání je posledním bodem, který je součástí semestrálního projektu. Čtvrtým bodem bakalářské práce je provedení vyhodnocení na vhodném příkladě a provedení zevrubné diskuze výsledků. Závěrečným pátým bodem zadání je zhodnocení dosažených výsledků v bakalářské práci a možnosti pokračování práce.

### 3.3 Návrh vyhodnocení úspěšnosti detekce

Řešení vyhodnocení úspěšnosti detekce objektů v obraze, které bylo navrhováno dosud neexistuje nebo není rozšířeno. Bylo nutné jej v rámci této práce vytvořit. V této části je popsán návrh a plán řešení vyhodnocení úspěšnosti detekce. Řešení je konkrétně popsáno v následující kapitole práce.

Základním předpokladem pro vyhodnocení úspěšnosti detekce objektů v obraze je ověření správnosti hodnot detekovaného objektu. Mezi tyto hodnoty patří souřadnice  $x$ , souřadnice  $y$ , šířka  $w$  a výška  $h$  detekovaného objektu. Hodnoty souřadnic, výšky a šířky definují, na které pozici v obraze se objekt nachází a jaká je jeho velikost. Je nutné nalézt způsob, jak ověřit správnost těchto hodnot detekovaného objektu, jež jsou výstupem detektoru. Pro ověření správnosti hodnot detekovaného objektu jsou nutné nějaké výchozí neboli referenční hodnoty (tzv. anotace), které by se porovnávaly s hodnotami detekovaného objektu. Správnost těchto hodnot musí být před detekcí objektů předem ověřena. Hodnoty, které mají ověřenou správnost, jsou nazývány jako referenční hodnoty nebo referenční souřadnice. Tyto pojmy jsou nadále používány v této bakalářské práci. Je to pravděpodobně jediný efektivní způsob, jak ověřovat správnost detekce. Zkrátka je nutné mít již před samotnou detekcí výstup z detektoru, jehož správnost je před detekcí ověřena. Je vhodné použít takové nastavení detektoru, které zaručuje nejvyšší možnou úspěšnost. Takové nastavení je zapotřebí, aby se nemuselo opravovat zbytečně příliš mnoho hodnot, ve kterých by se mohla potenciálně vyskytnout chyba.

Referenční hodnoty mohou mít různý formát. Nabízí se například formát prostého textu, XML, JSON, ASN.1 s použitím například BER kódování, TLV a další. Použití daného serializačního formátu záleží na osobních preferencích uživatele. Je nutné zvolit vhodný formát pro reprezentaci těchto hodnot, který zohledňuje výstupní formát detektoru. Důležité je také vyřešit způsob ukládání referenčních hodnot. Nabízí se dvě možné varianty pro jejich uložení. První variantou je uložení do externího souboru, který se později načte, zpracuje a vyhodnotí. Lze ještě zvolit, zda tyto hodnoty ukládat do jediného souboru nebo je ukládat pro každou fotografii do souboru zvlášť. Při ukládání zvlášť do souborů odpadá nutnost parsování jednotlivých hodnot, které by se muselo při použití ukládání do jediného souboru provádět. Druhou variantou je zahrnout tyto hodnoty jako součást programu. Tohle je však velmi nevhodné a nekonvenční řešení.

Referenční hodnoty a hodnoty detekovaného objektu se následně porovnávají. Porovnání je zohledněno nastavením nějaké rezervy neboli rozsahu, kdy je ještě detekce považována za úspěšnou. Rozsah úspěšnosti je použit, aby se jasně odlišila situace úspěšné a neúspěšné detekce. Tento rozsah je také zvolen, protože hodnoty detekovaného objektu nemusí přesně odpovídat referenčním hodnotám. Vlivem předzpracování fotografie může odchylka mezi hodnotami detekovaného objektu a referenčními hodnotami ještě narůst. Tuto skutečnost je potřeba také při implementaci zohlednit. Při zvětšování a zmenšování fotografie je nutné provádět přepočty referenčních hodnot pro správné porovnání s hodnotami detekovaného objektu. Způsob porovnávání a použití rozsahu úspěšnosti detekce je konkrétně popsán v následující kapitole.

Výstupem tohoto porovnávání je celkový počet objektů, počet úspěšně detekovaných objektů a počet neúspěšných detekcí. Z tohoto výstupu, který je zpracován v rámci běhu programu, se následně vypočítá úspěšnost detekce objektů. Postup výpočtu je podrobně rozepsán v následující kapitole popisující řešení.

Dále bylo nutné navrhnout způsob a formát výstupu úspěšnosti detekce objektů v obraze. Bylo vhodné zvolit formát vyjádření úspěšnosti detekce. Nabízí se procentuální vyjádření nebo pomocí vyjádření pravděpodobnosti. Také je potřeba navrhnout způsob uchování těchto dat pro následné zpracování a vygenerování grafů. Vhodné je ukládat úspěšnost a příslušná data do datového souboru, ze kterého se poté tato data pro generování grafů načtou.

Vyhodnocení úspěšnosti detekce lze provádět v rámci daného parametru experimentu nebo v rozsahu všech parametrů experimentu. Pro zjištění kvality vybraného detektoru je vhodné vyhodnotit úspěšnost všech parametrů experimentu. Výstupem tohoto vyhodnocení je informace, zda je detektor schopen detekovat objekt alespoň na jednom z parametrů experimentu. Tento způsob vyhodnocení je mírně odlišný. Nevyhodnotí se úspěšnost detekce jedné fotografie v rámci daného kroku experimentu, ale prochází se všechny detekované hodnoty fotografie v rámci celého experimentu. Porovnání s referenčními hodnotami probíhá stejným způsobem. Musí být navržen a implementován způsob uchování výsledků detekce. Při uchování výsledků detekce se musí počítat i s uložením úspěšnosti detekce více objektů. Pokud je detekce úspěšná při určitém parametru experimentu, tak se již hodnota následně nepřepisuje nezávisle na výsledcích dalších vyhodnocení. Hodnota úspěšnosti se počítá z hodnot zaznamenaných v použité datové struktuře.

## 3.4 Návrh experimentů

Nejprve je popsán návrh experimentů s parametry detektoru, které se zadávají standardně jako parametry programu v terminálu při spuštění detektoru. Mezi tyto parametry patří velikost kroku posunu detekčního okna, zvětšení, zmenšení, počet úrovní zmenšení a podobně. Pro moje experimenty je používán především parametr pro nastavení velikosti kroku posunu detekčního okna. Zvětšování a zmenšování fotografie je u detektoru při provádění některých experimentů vypnuto. Detektor provádí zvětšování a zmenšování pro některé experimenty nevhodným způsobem. V rámci některých experimentů se také provádí zvětšení a zmenšení zvoleným nástrojem pro předzpracování vstupní fotografie před detekcí objektů. Provádění změny měřítka vstupní fotografie je součástí experimentů s převzorkováním. Tyto experimenty bylo vhodné sloučit.

Další experimenty jsou založeny na předzpracování fotografie před detekcí objektů. Prvním takovým experimentem je provádění rotací vstupní fotografie. Tyto rotace provádí také v rámci předzpracování vhodný nástroj, který byl potřeba zvolit. Je vhodné provádět kladné i záporné rotace fotografií. Nutné je navrhnout vhodný rozsah testovaných hodnot a hodnotu kroku, které se použijí pro experiment.

Následující experiment provádí afinní transformace vstupní fotografie. Afinní transformace se provádějí také pomocí nástroje pro předzpracování. Pomocí afinní transformace lze provádět různé operace se vstupní fotografií. Je proto nutné navrhnout vhodný experiment, jehož výsledek bude přínosný pro uživatele a také pro ověření funkčnosti a správnosti algoritmu. Lze například provádět zkosení, snížení, zúžení nebo různé deformace a zkreslení vstupní fotografie.

Poslední skupina experimentů provádí změnu měřítka původní fotografie s použitím speciálních převzorkovacích filtrů a interpolačních metod. Tyto operace se provádějí také pomocí nástroje pro předzpracování vstupní fotografie. Nejprve se mění pouze měřítko vstupní fotografie bez použití dalších metod. Tato změna měřítka se provádí mimo detektor. V podstatě se provádí pouze zvětšování a zmenšování fotografie, které bylo již výše popsáno. Další experimenty již mění měřítko fotografie s použitím převzorkovacích filtrů a interpolačních metod. Změnu měřítka fotografie lze provádět několika způsoby. Je možné například měřítko měnit lineárně. To znamená, že velikost kroku změny měřítka se v průběhu experimentu nemění. Velikost kroku měřítka však lze v průběhu experimentu různě měnit. Je vhodné provést obě varianty experimentu se změnou měřítka. Existuje velké množství převzorkovacích filtrů a interpolačních metod, které lze aplikovat při změně měřítka. Výstup těchto filtrů a metod zaznamenává různé výsledky. Kvalita převzorkování výstupní fotografie může být značně odlišná. Bylo potřeba zvolit několik převzorkovacích filtrů a interpolačních metod pro experiment a zjistit vliv jejich použití na úspěšnost detekce objektů v obraze.

Po provedení experimentů se výsledné hodnoty, které jsou výstupem detektoru vybraným způsobem ukládají ve zvoleném formátu. Tyto soubory se poté načtou a jejich hodnoty se porovnávají s referenčními hodnotami, které byly získány předem a byla zkontrolována jejich správnost. Výstupem je úspěšnost detekce dané fotografie. Tento výstup je ukládán do datového souboru. Z těchto datových souborů jsou nakonec vyneseny grafy, ze kterých uživatel snadno získá efektivní nastavení pro detektor a optimální předzpracování fotografií. Konkrétní popis řešení je uveden v následující kapitole.

## 3.5 Požadavky na řešení

Před začátkem práce bylo nutné zvolit prostředky, které jsou použity k vypracování zadaných úkolů. Cílem bylo vytvořit skript, který spouští vybrané experimenty a vyhodnocuje úspěšnost detekce objektů. Vhodné tedy bylo použít nějaký skriptovací programovací jazyk. Nabízí se například použití skriptovacího programovacího jazyka typu Shell nebo jazyka Python. Zvolen byl nakonec první ze dvou zmíněných, konkrétně Bash Shell z důvodu předchozí znalosti, jednodušší práce se soubory a snadným spuštěním dalších potřebných programů a nástrojů. Nevýhodou je absence knihoven, kterými naopak bohatě disponuje skriptovací jazyk Python.

Použit byl WaldBoost detektor s LBP příznaky, který se spouští z příkazové řádky terminálu. Vzhledem k použití tohoto detektoru bylo vhodné použít skriptovací jazyk typu Shell, protože Python není vhodný ke spuštění dalších programů a nástrojů.

Výstup detektoru je realizován na standardní výstup v serializačním formátu JSON. Výstup lze tudíž jednoduše ukládat do externího souboru. Z tohoto důvodu byl potřeba program pro parsování a filtrování dat ze souboru ve formátu JSON. Vhodným kandidátem byl program jq, který je jednoduchý a velmi efektivní.

Pro předzpracování fotografií pro detektor byl potřeba nástroj, který dovede provádět zvětšování, zmenšování, rotaci, afinní transformaci, převzorkování, ořezání fotografie, aplikovat různé filtry a interpolace při převzorkování. Vhodným nástrojem, který je hodně rozšířený by byla knihovna OpenCV. Problém je, že ji nelze použít přímo z příkazové řádky terminálu. Nutností je C/C++ nebo Python a další. Pro předzpracování fotografií byl tedy zvolen nástroj convert z balíku ImageMagick. Tento nástroj je velmi efektivní a poskytuje velké množství operací pro úpravu, transformace, převzorkování a podobně. Výhodou je podpora kvalitního převzorkování s použitím filtrů a interpolací. Výhodou tohoto nástroje také je, že lze jednoduše spouštět z terminálu. Tudíž splňuje všechny požadavky.

Výsledky experimentů je vhodné zobrazit přehledně pomocí grafů, aby mohl uživatel intuitivně nalézt efektivní parametry detektoru a předzpracování fotografií pro detektor obličejů. Tyto grafy jsou vytvářeny z datových souborů, jež jsou výstupem skriptu a naplněny daty z výsledků experimentů. K vynášení grafů byl použit nástroj Gnuplot, který umožňuje prokládání vynesných bodů různými křivkami a polynomy. Tento nástroj disponuje širokým nastavením a navíc výsledné grafy jsou ostré a přehledné. Samozřejmostí jsou i 3D grafy.

Pro operace s pohyblivou desetinnou čárkou byl použit jednoduchý, ale přesto velmi efektivní kalkulátor bc, jež je předinstalovaným nástrojem linuxových operačních systémů. Tento kalkulátor byl použit pro efektivní výpočet úspěšnosti detekce. Bash shell totiž neumožňuje operace s pohyblivou desetinnou čárkou.



## 4 Popis práce a zhodnocení výsledků

V tomto stavu je vybrán software, který je použit k řešení práce. Zvolen je také programovací jazyk, ve kterém je algoritmus vyhodnocení úspěšnosti detekce objektů implementován. Dále je vybrán dobře parametrizovatelný detektor obličejů. Připraven je vhodný nástroj pro předzpracování (transformaci, rotaci, zvětšování, převzorkování a ořezávání) vstupních fotografií před detekcí. Jsou vybrány postupy a metody pro splnění cíle bakalářské práce. Řešení vyhodnocení úspěšnosti detekce objektů v obraze je vymyšleno. Jsou taktéž navrženy experimenty s nastavením detektoru a předzpracováním vstupních fotografií, které je prováděno před detekcí objektů. Tato kapitola se detailně zabývá řešením bakalářské práce. Je popsáno řešení navrženého vyhodnocení úspěšnosti detekce, jednotlivé experimenty s parametry detektoru a předzpracováním fotografií. Na závěr kapitoly jsou zhodnoceny dosažené výsledky této práce.

### 4.1 Popis experimentů

Pro ověření funkcionality a především správnosti algoritmu vyhodnocení úspěšnosti detekce objektů v obraze vznikly experimenty, které provádějí pokusy s různým nastavením detektoru a předzpracováním vstupních fotografií. Tyto experimenty také slouží pro nalezení optimálního nastavení detektoru pro detekci objektů a optimálního předzpracování vstupních fotografií před detekcí. Byly vytvořeny experimenty s krokem posunu detekčního okna, rotací, afinní transformací a změnou měřítka s různými způsoby převzorkování.

### Krok posunu detekčního okna

Detekční okno je jednoduše řečeno rámeček, který ohraničuje nějakou oblast na fotografii. Tento rámeček neboli okno se používá k vyhledání objektů na fotografii. Tohle okno prochází postupně celou fotografií a testuje se, zda se na dané pozici nachází hledaný objekt. Schopnost detekovat určitý objekt, konkrétně třeba obličej, je dána trénováním detektoru. Vhodná velikost tohoto okna pro detekci obličejů je přibližně 24x24 pixelů [3].

Velikost posunu tohoto okna značně ovlivňuje úspěšnost detekce. Čím menší krok posunu detekčního okna, tím je větší pravděpodobnost úspěšné detekce objektů v obraze. S použitím většího kroku se však urychluje průchod detekčního okna fotografií a tím se urychluje i celý proces detekce. Lze tedy očekávat, že se zvětšujícím se krokem posunu detekčního okna bude úspěšnost detekce objektů v obraze postupně klesat. Je proto dobré najít kompromis mezi rychlostí a úspěšností detekce, čímž se zvýší efektivita detektoru.

V rámci experimentu s krokem posunu detekčního okna se testují hodnoty posunu od 1 do 10 pixelů. Příklad spuštění detektoru s nastavením určitého kroku posunu je uveden níže.

```
./lbpdetector -d LBPDetector/data/frontal_faces_lbp.bin --stride=1  
img_001.jpg
```

### Rotace

Obličej na fotografiích mohou být různým směrem a o určitý úhel natočeny nebo dokonce přetočeny. Toto natočení znemožňuje úspěšně detekovat objekty na fotografii. V experimentu s rotací nebudou uvažovány výrazně přetočené fotografie. Předpokladem je nepravděpodobnost, že by uživatel pořizoval přetočené fotografie nebo videozáznam o -90 a méně stupňů nebo naopak o 90 a více stupňů. Pokud je to nutné například z technických důvodů, tak může uživatel takové fotografie nebo záznam jednoduše pro detektor přetočit. Testování i těchto větších úhlů rotace by zbytečně

prodlužovalo délku detekce objektů v obraze. Takových situací by při tvorbě datasetu mohlo nastat minimum nebo by nemusela nastat žádná taková situace.

Takto provedený experiment nemá pro samotnou detekci velký význam. Slouží spíše pro informaci o tom, jakou rotaci má smysl pro detekci objektů provádět. Zajímavé řešení by bylo implementovat rotaci v rámci detektoru. Efektivní by bylo použít hrubější krok při výběru parametrů rotace. Nevýhodou by bylo pravděpodobně znatelné zpomalení detekce. Takže pro použití při detekci v reálném čase toto řešení nebo vylepšení nemá pravděpodobně význam. Příklad rotace s použitím nástroje convert je uveden níže [20]. Nově vznikající pixely jsou definovány parametrem background, jehož výchozí barva je bílá. Krok mezi jednotlivými iteracemi experimentu s rotací je nastaven na 5°. K rotaci dochází v rámci předzpracování fotografie. Rozsah parametrů pro testování rotace byl nastaven od -40° do +40°. Pro tento experiment byly obličejové fotografie ořezány navíc s 25 pixely na každé straně fotografie [14].

```
convert -rotate 30 img_001.jpg img_001_30.jpg
```

## Afinní transformace

Afinní transformace je velmi zajímavá a mocná operace, kterou lze použít na fotografii. Je možné pomocí ní provádět rotaci, zkosení, posunutí, změnu měřítka a kombinace těchto operací. Výsledkem může být různé zkreslení původní fotografie. Výsledná transformace fotografie je dána šesti parametry transformační matice, které určují výslednou pozici daného pixelu na fotografii. Afinní transformace je v rámci předzpracování realizována pomocí funkce affine, jejíž parametry jsou  $s_x$ ,  $r_x$ ,  $r_y$ ,  $s_y$ ,  $t_x$ ,  $t_y$  a pomocí funkce transform nástroje convert [16]. Tato operace získá jako vstup souřadnice určitého bodu respektive pixelu fotografie a ten je následně transformován pomocí transformační matice. Tvar transformační matice je uveden níže. Souřadnice  $x'$  a  $y'$  jsou souřadnice nového bodu transformované fotografie. Tento bod je výsledkem vynásobení původních souřadnic  $x$ ,  $y$  a transformační matice. Tato operace násobení matic se dá zjednodušit na níže uvedenou rovnici [16].

$$\begin{vmatrix} x' & y' & 1 \end{vmatrix} = \begin{vmatrix} x & y & 1 \end{vmatrix} \cdot \begin{vmatrix} s_x & r_x & 0 \\ r_y & s_y & 0 \\ t_x & t_y & 1 \end{vmatrix}$$
$$(x', y') = (x \cdot s_x + y \cdot r_y + t_x, x \cdot r_x + y \cdot s_y + t_y)$$

Po dosažení níže uvedených parametrů afinní transformace je výstupem původní fotografie. To znamená, že žádná transformace neproběhne [16].

$$s_x = 1, r_x = 0, r_y = 0, s_y = 1, t_x = 0, t_y = 0$$

V rámci experimentů prováděných za účelem zjištění efektivního předzpracování fotografie před detekcí objektů je prováděno pomocí afinní transformace pouze zkosení původní fotografie. Toto zkosení je prováděno na ose  $x$  a na ose  $y$ . Kombinace těchto operací nebudou prováděny, i když výsledek této transformace by byl zajímavý. Dosazuje se tedy pouze za parametry  $r_x$  a  $r_y$ . Rozsah testovaných parametrů je od -0.7 do 0.7 za předpokladu transformace pouze detekovaného objektu. Přesněji řečeno, detekce samotného již detekovaného objektu byla ve většině případů neúspěšná. Z toho důvodu každá strana detekovaného objektu byla ořezána navíc s 25 pixely [14]. Díky tomuto řešení došlo ke zrychlení experimentů s afinní transformací a především lze takto jednodušeji zjišťovat úspěšnost detekce. Při porovnávání JSON struktury detekovaného objektu s referenční strukturou docházelo k velkým změnám detekovaného rámečku. Z tohoto důvodu vznikaly obtíže se zjišťováním úspěšnosti detekce, která byla v některých případech zkreslená. Z tohoto důvodu byly

referenční souřadnice x a y nastaveny na velikost, o kterou byl zvětšen ořezaný objekt. Následné porovnání struktur detekovaných objektů s těmito referenčními strukturami bylo úspěšné a rozumné. Krok experimentu zkosení pomocí afinní transformace je nastaven na hodnotu 0.1. Příklad použití afinní transformace nástrojem convert je uveden níže. Parametr matte slouží pro nastavení matné průhledné hodnoty pixelu. Parametrem virtual-pixel se nastavuje způsob vyplnění pixelů, které se nacházejí mimo původní obraz. Jsou to virtuální pixely, které vznikly při transformaci vstupní fotografie. Argument Transparent znamená nastavení hodnoty pixelů na černou průhlednou barvu. Parametr repage aktualizuje hodnoty souřadnic rámečku fotografie a její metadata [16].

```
convert img_001.jpg -matte -virtual-pixel Transparent -affine
1,0,0,1,0,0 -transform +repage img_001_100100.jpg
```

## Převzorkování

Převzorkování původní fotografie v rámci předzpracování fotografie před detekcí objektů v obraze může snížit dobu trvání detekce. Snížení doby trvání detekce nastává za předpokladu, že dochází ke snižování vzorkovací frekvence. Součástí mého experimentu je při provádění převzorkování jak snižování, tak i zvyšování vzorkovací frekvence. To v praxi znamená slučování pixelů nebo naopak vytváření nových pixelů. Výsledkem je tedy změna struktury pixelů původní fotografie. Existuje spousta metod pro výpočet nových hodnot pixelů při jejich slučování a rozdělování. Každá z těchto metod dosahuje různých výsledků a nemusí znamenat, že použití složitější metody přinese nejlepší výsledek. Na kvalitu fotografie má použitá metoda převzorkování velký vliv. Na úspěšnost detekce objektů v obraze již použitá metoda převzorkování nemá tak výrazný vliv [17].

Mezi nejpoužívanější metody k převzorkování obrazu patří interpolace nebo různé speciálních převzorkovacích filtry. Obvykle při použití interpolace se používá bilineární interpolace, bikubická interpolace, interpolace pomocí splinů, průměru, nejbližšího souseda, sítě a další. Mezi používané filtry patří například Lanczos, Lagrange, Catrom, Hamming, Parzen, Blackman, Kaiser, Welsh, Hanning, Bartlett, Bohman a další [17] [18].

Pro experimenty byla použita bilineární a bikubická interpolace. V případě převzorkovacích filtrů byl použit Lanczosův filtr, Lagrangeův filtr a Catmull-Rom filtr. Zmenšování a zvětšování se provádí procentuálně z důvodu srozumitelnosti a informační hodnotě pro uživatele. Změna měřítka prováděna po pixelech by neměla význam a pro uživatele by byla nic neříkající. V experimentu se tedy používají pro snižování a zvyšování vzorkovací frekvence hodnoty od 10% do 300%. Krok mezi hodnotami experimentu je 10%, 20%, 1/4 nebo 1/6, dle zvoleného experimentu.

Díky procentuální změně měřítka a zobrazení výsledku lze velmi jednoduše a efektivně přizpůsobit algoritmus pro porovnávání struktur při vyhodnocení úspěšnosti. Stačí pouze procentuální hodnotu změny měřítka vydělit 100. Poté touto získanou hodnotou vynásobit referenční souřadnice, výšku a šířku. Tímto výpočtem získané hodnoty odpovídají hodnotám detekovaného objektu. Stačí nastavit rozsah úspěšnosti detekce. Empiricky zjištěná optimální hodnota rezervy je od -15 do +15 pixelů u souřadnic a od -20 do +20 pixelů u výšky a šířky detekovaného objektu. Zjišťování úspěšnosti detekce objektů tímto způsobem je vysoce efektivní a přesné.

Bylo vytvořeno 13 experimentů se změnou vzorkování. Tyto experimenty provádí změnu měřítka s různými způsoby převzorkování. Každý z těchto experimentů používá odlišnou metodiku. Používají se interpolační metody a převzorkovací filtry, které jsou vyjmenovány výše a blíže popsány níže. U některých experimentů se před spuštěním využívá předzpracování obrazu. Taktéž se provádí zpracování obrazu mezi jednotlivými kroky některých experimentů.

První experiment provádí změnu měřítka s různými způsoby převzorkování. Provádí se pravidelný krok v jednotlivých iteracích experimentu. Používá se celá nepředzpracovaná fotografie. Krok experimentu je nastaven na 10%. V každém kroku se používá původní fotografie. Experiment provádí zmenšování a zvětšování od 10% do 300%.

Druhý experiment provádí změnu měřítka s různými způsoby převzorkování. Provádí se také pravidelný krok. Na rozdíl od předchozího experimentu používá předzpracované fotografie, na kterých jsou ořezány obličejové tváře navíc s 25 pixely. Krok experimentu je nastaven na 10%. V každém kroku se používá původní (předzpracovaná) fotografie. Experiment provádí taktéž zmenšování a zvětšování od 10% do 300%.

Třetí experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Provádí se zmenšování z původní velikosti fotografie. Krok experimentu je nastaven na 20%. Provádí se postupně 6 kroků zmenšení.

Čtvrtý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Nejprve je původní obraz zmenšen na 25% a poté je postupně zvětšován po 20%. V experimentu se provádí 8 kroků zvětšení.

Pátý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Vstupem jsou celé nepředzpracované fotografie. Zmenšuje se na 3/4. Provádí se postupně 6 kroků zmenšení. Zmenšování na 3/4 je v krocích 3 až 6 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů.

Šestý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Nejprve se předzpracují vstupní fotografie. Obličejové tváře jsou ořezány navíc s 25 pixely. V experimentu se nejprve provede zvětšení na 133%. Poté se zmenšuje na 3/4. Postupně je použito 6 kroků zmenšení. Zmenšování na 3/4 je v krocích 3 až 6 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů.

Sedmý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Obličejové tváře jsou ořezány navíc s 10 pixely. Poté jsou zmenšeny na velikost 31x31 pixelů. V experimentu se provádí zvětšování na 133% v každém kroku. Poté se zmenšuje na 3/4. Je použito 11 kroků. Zmenšování na 3/4 je v krocích 3 až 11 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů. Při tomto experimentu je u detektoru vypnuto zmenšování a jsou sníženy limity velikosti fotografie.

Osmý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Obličejové tváře jsou ořezány navíc s 10 pixely. Poté jsou zmenšeny na velikost 31x31 pixelů. V experimentu se provádí zvětšování na 133% v každém kroku. Zvětšování se provádí z původní fotografie najednou. Poté se zmenšuje na 3/4. Postupně je použito 11 kroků. Zmenšování na 3/4 je v krocích 3 až 11 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů. Při tomto experimentu je u detektoru vypnuto zmenšování a jsou sníženy limity velikosti fotografie.

Devátý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Vstupem jsou celé fotografie. Postupně se provádí zmenšování na 5/6. Použito je 7 kroků zmenšení. Zmenšování na 5/6 je v krocích 4 až 7 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů.

Desátý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskádu převzorkování. Na vstupních fotografiích se ořezávají obličejové tváře navíc s 25 pixely. Nejprve se provede zvětšení na 120%. Poté se postupně zmenšuje na 5/6. Provádí se 7 kroků zmenšení. Zmenšování na 5/6 je v krocích 4 až 7 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů.

Jedenáctý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Obličejové tváře jsou ořezány navíc s 10 pixely. Poté jsou zmenšeny na velikost 31x31 pixelů. V experimentu se provádí zvětšování na 120% v každém kroku. Poté se zmenšuje na 5/6. Postupně je použito 11 kroků. Zmenšování na 5/6 je v krocích 4 až 11 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů. Při tomto experimentu je u detektoru vypnuto zmenšování a jsou sníženy limity velikosti fotografie.

Dvanáctý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Obličejové tváře jsou ořezány navíc s 10 pixely. Poté jsou zmenšeny na velikost 31x31 pixelů. V experimentu se provádí zvětšování na 120% v každém kroku. Zvětšování probíhá z původní fotografie najednou. Poté se zmenšuje na 5/6. Postupně je použito 11 kroků. Zmenšování na 5/6 je v krocích 4 až 11 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo

jádru a redukce řádků a sloupců pixelů. U detektoru je vypnuto zmenšování a jsou sníženy limity velikosti fotografie.

Poslední třináctý experiment provádí změnu měřítka s různými způsoby převzorkování. Používá se kaskáda převzorkování. Vstupem jsou celé fotografie. Fotografie se postupně zmenšují na 5/6. Provádí se postupně 15 kroků zmenšení. Zmenšování na 5/6 je v krocích 4 až 15 nahrazeno zmenšováním na 1/2. U zmenšování na 1/2 je použito Gaussovo jádro a redukce řádků a sloupců pixelů. Na rozdíl od ostatních experimentů se vyhodnocuje úspěšnost detekce celé kaskády.

## Zmenšení a zvětšení

Zmenšování a zvětšování fotografie se u detekce objektů v obraze využívá z důvodu možné nevhodnosti rozměrů detekovaného objektu. Objekt, který se detektor pokouší detekovat, může být příliš malý nebo naopak příliš velký. Díky této skutečnosti může být pro detektor obtížné objekt detekovat. Zmenšení nebo zvětšení lze na fotografii aplikovat před detekcí v rámci předzpracování fotografie nebo tuto změnu měřítka může provádět i detektor. Pro experimenty bylo nutné provádět zmenšování a zvětšování v rámci předzpracování fotografie. Bylo nutné provádět toto zmenšování a zvětšování v rozumné míře a s vhodným krokem. U některých experimentů nebylo optimální použít zmenšování a zvětšování pomocí detektoru, který prováděl změnu měřítka pro vybrané experimenty nevhodným způsobem. Lze provádět zmenšování a zvětšování na hodnotu zadanou v pixelech nebo procentech. Pro přehlednost výsledného grafu je vhodnější použít procentuální zmenšování respektive zvětšování a také procentuální zobrazení výsledného zmenšení a zvětšení v grafu. Tento způsob zmenšování a zvětšování je přehledný pro uživatele, který si tak snadno dokáže představit výsledný efekt na fotografii. Zmenšování a zvětšování fotografie po pixelech je nevhodné a uživateli, kterému není známo rozlišení původní fotografie, nepřináší požadovanou informační hodnotu. V experimentu se změna měřítka provádí pomocí nástroje convert dle níže uvedeného příkladu [15]. Při změně měřítka fotografie se používá replikace pixelů [17].

```
convert img_001.jpg -filter point -distort resize 100%  
img_001_100.jpg
```

## Lanczosův filtr

Tento filtr je jeden z neznámějších a zároveň nepoužívanějších filtrů existujících k převzorkování obrazu. Nazván je po maďarském matematikovi a fyzikovi Corneliu Lanczosovi. Lanczosův filtr patří mezi filtry, které používají tzv. okno. Nepatří zrovna mezi rychlé filtry, ale ani nepatří k pomalým filtrům. Není ani nejlepší, ale poskytuje solidní výsledky. Tento filtr není příliš složitý a je tedy vhodný pro začátek práce s kvalitnějším převzorkováním. Lanczosův filtr v základní formě využívá jeden hlavní (centrální) lalok funkce kardinálního sinu, respektive jeho pozitivní část. V experimentech je používána verze Lanczosova filtru, která využívá tři laloky funkce kardinálního sinu. Použití více laloků je vhodné, protože funkce okna při použití jednoho laloku tento lalok ořeže a zbytek funkce kardinálního sinu je nevyužitý. Nevýhodou použití jednoho laloku je také možnost vzniku artefaktů při překmitu, který může generovat funkce okna kardinálního sinu [7] [17].

Převzorkování s pomocí Lanczosova filtru se provádí pomocí nástroje convert z balíku ImageMagick. Zápis konkrétního příkazu je uveden níže [17].

```
convert img_001.jpg -filter lanczos -distort resize 200%  
img_001_200.jpg
```

Výchozí hodnota počtu laloků u Lanczosova filtru je 3 laloky. Tuto hodnotu lze však nastavit na požadovaný počet laloků pomocí příkazu define filter, který je uveden níže [17].

```
convert img_001.jpg -filter lanczos -define filter:lobes=2 -distort
resize 200% img_001_200.jpg
```

## Lagrangeův filtr

Jedná se o složitější filtr od pana Josepha Louise Lagrangeho. Matematicky se jedná o velmi pomalý filtr. Výpočetně srovnatelný s Gaussovým filtrem. Používá funkci okna, kardinální sinus a sombrero funkci. Sombrero funkce je v podstatě 2D analogická funkce kardinálního sinu s rozdílem, že používá polární souřadnice. Z důvodu větší komplexnosti výpočtu je tudíž velmi pomalá, protože počítá trigonometrické funkce pro váhy a funkce okna. Příklad použití Lagrangeova filtru je uveden níže. Taktéž je použit nástroj convert a jeho funkce filter a distort [17].

```
convert img_001.jpg -filter lagrange -distort resize 200%
img_001_200.jpg
```

Lagrangeův filtr generuje po částech funkci kubického polynomu pro aproximaci filtru okna. Pomocí parametru support lze nastavit podporující okno. Je to jakési klouzající okno, které počítá průměr a zvětšuje původní obraz. Pokud je hodnota například nastavena na 1.25, oblast podpory je 2.5 pixelů. Příklad použití podpory je uveden níže [17].

```
convert img_001.jpg -filter lagrange -define filter:support=1.25
-distort resize 200% img_001_200.jpg
```

Problém je, že při použití podpory může vznikat aliasing a zkreslení. S větší hodnotou podpory Lagrangeův filtr nepotřebuje počítat komplexní trigonometrické funkce nebo další funkce okna. Použitím větší hodnoty podpory se emuluje funkce kardinálního sinu [17]. Podpora filtru je z matematického hlediska velmi výhodná. Získané výsledky jsou přesnější. V praxi je však použití podpory omezené [17]. Podporu lze použít i u dalších filtrů (Gaussův filtr, trojúhelníkový filtr).

## Catrom filtr

Catmull-Rom je přesný název tohoto filtru. Je pojmenován po Edwinu Catmullovi a Raphaelu Romovi, kteří jej vytvořili. Je to velmi známý kubický filtr používaný k interpolaci. Produkuje rozumně ostré hrany a generuje hladký a nerozmazaný obraz. Může však docházet k překmitu a blokování. U významnějšího zvětšení nemusí být zřetelný gradient. Tento kubický filtr je mnohem rychlejší a matematicky méně náročnější na generování. Vytváří kompromis mezi překmitem a blokováním. Snaží se najít nejlepší možný vyrovnaný výsledek z hlediska vznikajících artefaktů. Velmi zajímavé je, že je téměř stejný jako Lanczosův filtr s použitím dvou laloků, respektive poskytuje téměř duplikátní výsledek. Catrom filtr lze použít pomocí níže uvedeného příkazu. Použije se k tomu již známá funkce filter a distort nástroje convert [17].

```
convert img_001.jpg -filter catrom -distort resize 200%
img_001_200.jpg
```

## Bilineární interpolace

Bilineární interpolace je základní interpolační metoda. Je to jeden z nejjednodušších způsobů získání interpolovaného výsledku. Je to metoda, která je rozšířením metody lineární interpolace. Nejprve se provádí lineární interpolace ve směru osy x a poté ve směru osy y. Provádí se interpolace funkce dvou proměnných na pravidelnou prostorovou mřížku. Tato interpolace vyžívá kombinování barev pixelů okolo vzorkovacího bodu neboli pixelu. Jednoduše spojuje přímoú čarou v

ortogonálních směrech 4 sousední pixely pro získání barvy daného pixelu. Získaný výsledek je ekvivalentní trojúhelníkovému vzorkovacímu filtru. Příklad použití bilineární interpolace pomocí nástroje convert a jeho funkce interpolate je uveden níže [17] [18].

```
convert img_001.jpg -interpolate bilinear -distort resize 200%  
img_001_200.jpg
```

## Bikubická interpolace

Stejně jako u předchozí metody, tak i bikubická interpolace je rozšířením kubické interpolace. Provádí se interpolace bodů na prostorovou pravidelnou mřížku. Získaný interpolovaný povrch je hladší, než povrchy získané bilineární interpolací a obsahuje méně interpolačních artefaktů. Bikubická interpolace se provádí interpolací 16 okolních bodů (4x4). Tato metoda je matematicky náročnější na výpočet a tudíž je díky tomu i pomalejší. V případě, že nezáleží na rychlosti výpočtu je bikubická interpolace upřednostňována před bilineární interpolací. Pro výpočet bikubické interpolace lze využít Langrangeových polynomů, kubických splinů nebo algoritmus kubické konvoluce. Příklad použití bikubické interpolace pomocí nástroje convert a funkce interpolate je uveden níže [17] [18].

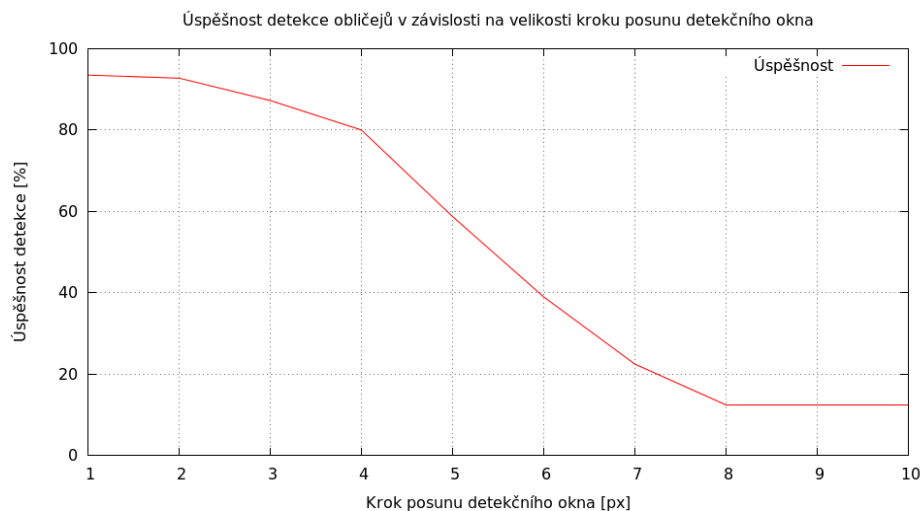
```
convert img_001.jpg -interpolate bicubic -distort resize 200%  
img_001_200.jpg
```

## 4.2 Výsledky experimentů

Tato podkapitola se zabývá popisem získaných výsledků, které jsou výstupem experimentů. Tyto experimenty byly výše konkrétně popsány. Interpretaci těchto výsledků se poté věnuje podkapitola, která provádí zhodnocení výsledků experimentů.

### Krok posunu detekčního okna

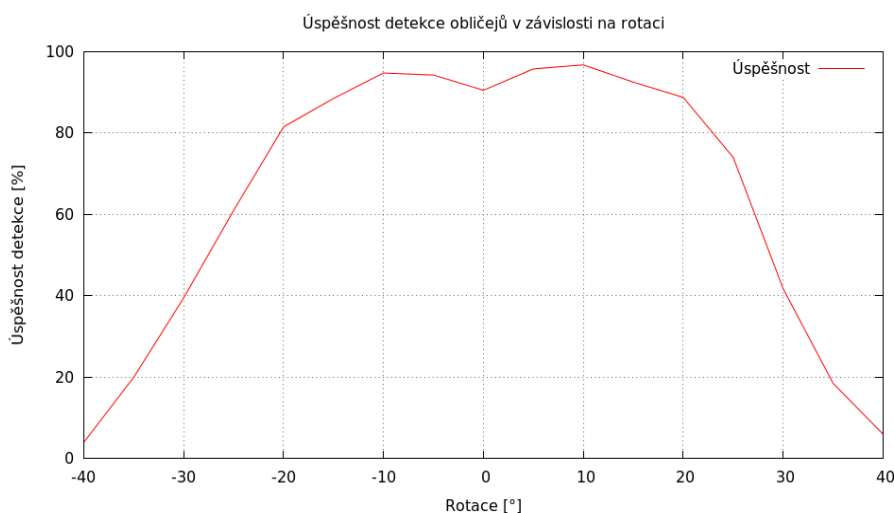
Prováděním testů na použitém datovém setu bylo zjištěno, že nemá smysl provádět experimenty s krokem posunu okna větším než 10 pixelů. Úspěšnost při tomto nastavení se blíží nule. Aktuálně je nastaveno v experimentu testovat kroky posunu detekčního okna o 1 až 10 pixelů, což je vzhledem k předpokládanému rozsahu experimentu dostačující. Do kroku posunu detekčního okna o 4 pixely je úspěšnost nad 80%. Od kroku posunu 5 pixelů úspěšnost zřetelně klesá, vizte obr. 4.2.1.



Obr. 4.2.1: Graf úspěšnosti detekce obličejů v závislosti na velikosti kroku posunu detekčního okna

## Rotace

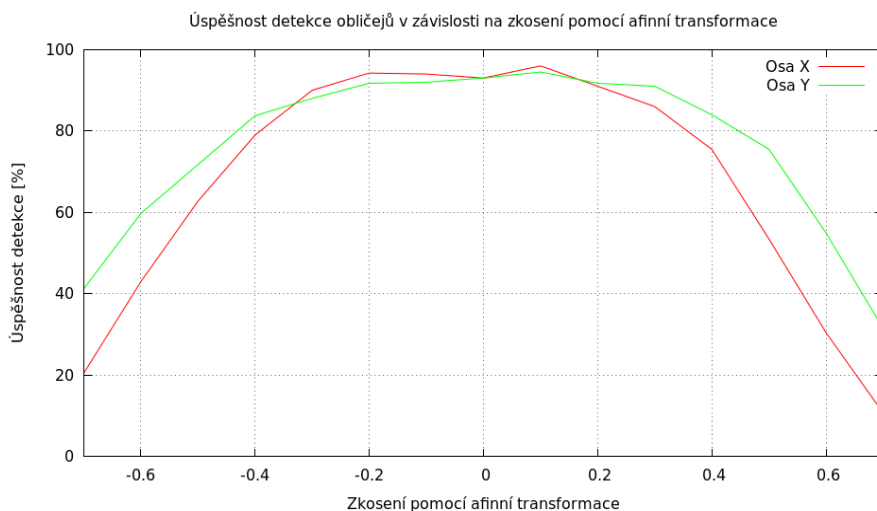
Detektor zaznamenal překvapivě úspěšnější výsledky, než bylo odhadováno. Za předpokladu, že je fotografie kvalitní a obličej se nachází přibližně v ose záběru fotoaparátu nebo kamery, lze dosáhnout úspěchu 80% a více přibližně od  $-15^\circ$  až ke  $20^\circ$  natočení. Je samozřejmé, že pořízené fotografie nejsou ve většině případů ideální. K rotaci dochází v rámci předzpracování fotografie. Krok mezi jednotlivými iteracemi experimentu s rotací je nastaven na  $5^\circ$ . Rozsah parametrů pro testování rotace byl nastaven od  $-40^\circ$  do  $+40^\circ$ . Menší nebo větší hodnoty než je uvedený rozsah nemají význam, protože úspěšnost detekce při použití těchto parametrů se blíží nule, vizte obr. 4.2.2.



Obr. 4.2.2: Graf úspěšnosti detekce obličejů v závislosti na rotaci

## Afinní transformace

Krok zkosení pomocí afinní transformace je nastaven na hodnotu 0.1, což se zdá být na první pohled hrubý krok. Z hlediska potřeby většího rozsahu hodnot se ve výsledku experimentu hodnota tohoto kroku projevila jako rozumně zvolená. Empiricky bylo zjištěno, že vhodný rozsah parametrů je od  $-0.7$  do  $0.7$  za předpokladu transformace pouze detekovaného objektu. Zajímavá je asymetrie kladné a záporné části funkce u obou testovaných dimenzí. Hodnoty parametrů od  $-0.4$  do  $0.3$  u osy x dosahují úspěšnosti nad 80%. Parametry od  $-0.4$  do  $0.4$  u osy y dosahují úspěšnosti nad 80%. Ostatní parametry experimentu již zaznamenaly prudší pokles úspěšnosti detekce, vizte obr. 4.2.3.

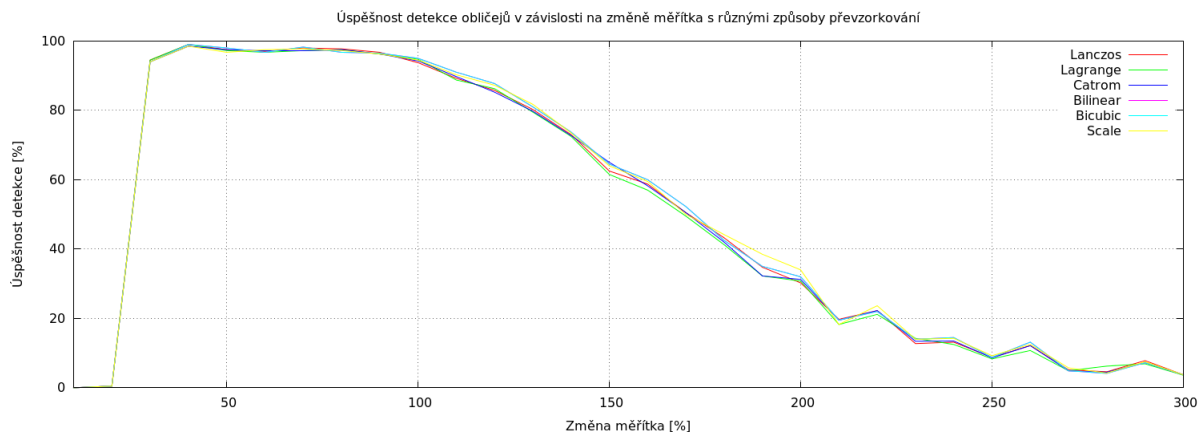


Obr. 4.2.3: Graf úspěšnosti detekce obličejů v závislosti na afinní transformaci



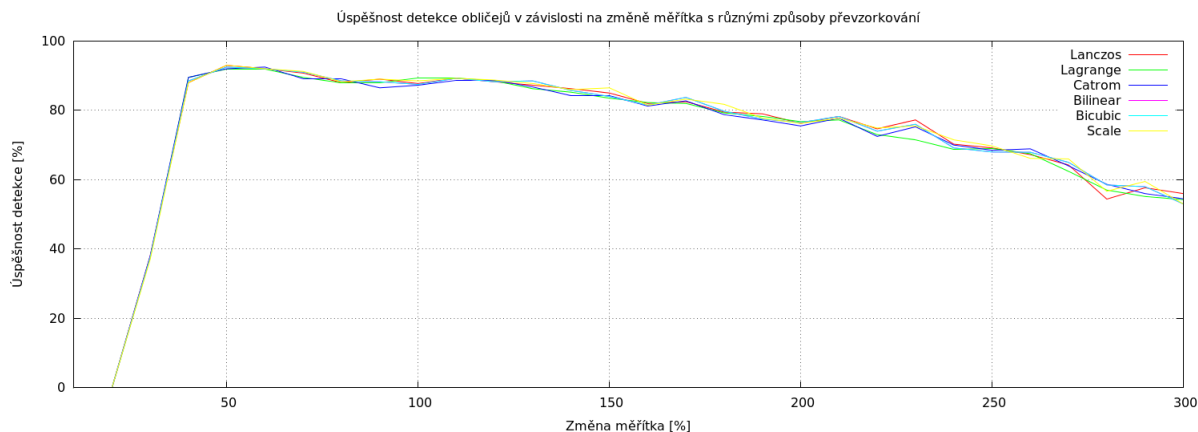
# Převzorkování

První experiment používá pravidelný krok mezi použitými parametry experimentu. V každé iteraci experimentu je brána vždy původní fotografie. Experiment je nastaven k postupnému zmenšování a zvětšování po 10%. Rozsah testovaných hodnot je od 10% do 300%. S významnějším zvětšením se úspěšnost detekce již blížíla k nule. Význam měl experiment od 25% do 125% zmenšení respektive zvětšení, kde se úspěšnost detekce pohybovala nad 80%. Menší hodnoty než 25% zaznamenaly velmi prudký propad úspěšnosti. Důvodem prudkého propadu úspěšnosti je limit velikosti fotografie u detektoru na 40x40 pixelů. U zvětšení vyššího než 120% již také úspěšnost klesala, nijak však dramaticky, vizte obr. 4.2.4. Převzorkovací filtry zaznamenaly srovnatelný výsledek jako interpolační metody a replikace pixelů.



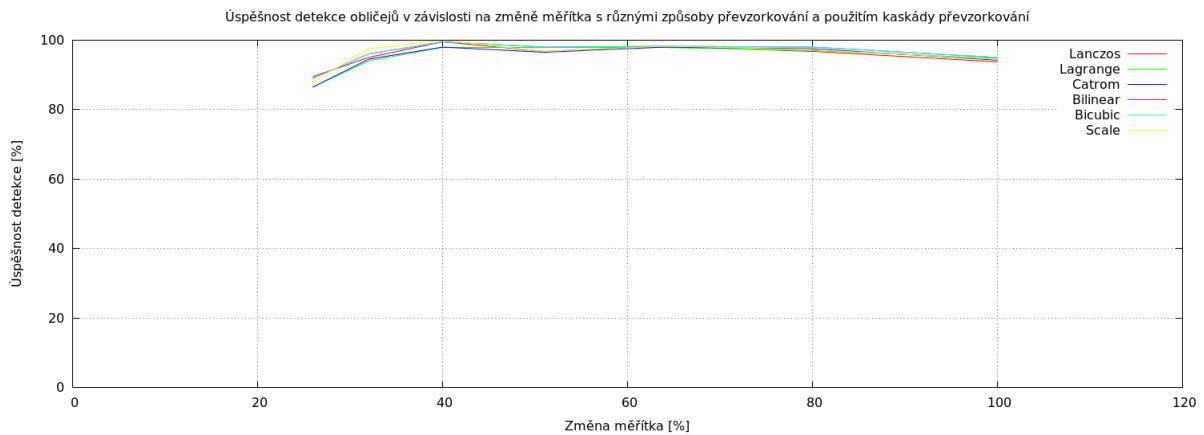
Obr. 4.2.4: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (pravidelný krok, vždy původní fotografie, zmenšování a zvětšování po 10%)

Druhý experiment používá pravidelný krok mezi jednotlivými použitými parametry experimentu. V každé iteraci experimentu je brána vždy původní fotografie. Nejprve se však na rozdíl od předchozího experimentu provádí ořez obličejů navíc s 25 pixely. Experiment je nastaven k postupnému zmenšování a zvětšování po 10%. Rozsah testování je od 10% do 300% zmenšení respektive zvětšení. U ořezaných obličejů se úspěšnost i s významnějším zvětšením držela nad 60%. Význam měl experiment od 25% do přibližně 200% zmenšení nebo zvětšení, kdy se úspěšnost detekce pohybovala nad 80%. Menší hodnoty než 25% také zaznamenaly prudký propad úspěšnosti, vizte obr. 4.2.5. Převzorkovací filtry zaznamenaly také u toho experimentu srovnatelný výsledek jako interpolační metody a replikace pixelů.



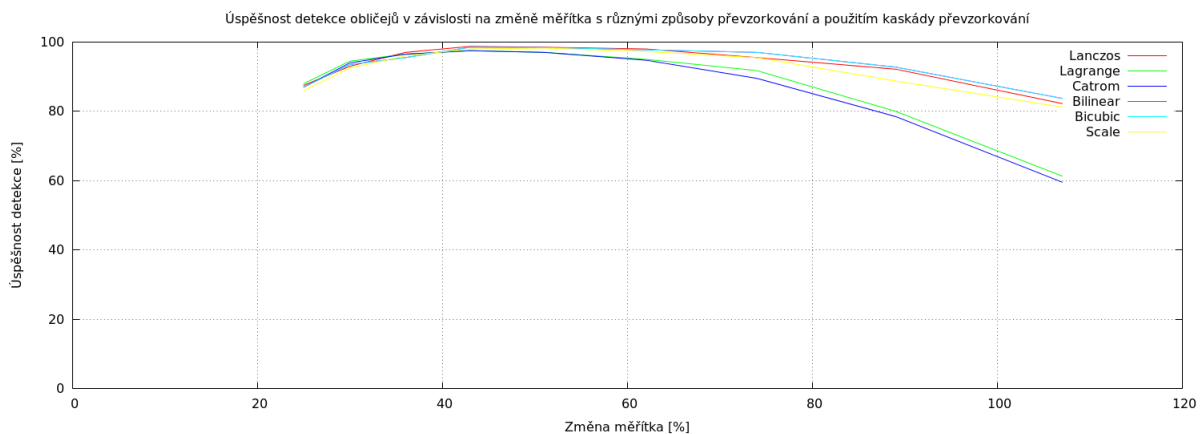
Obr. 4.2.5: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (pravidelný krok, ořez obličejů, v každém kroku původní fotografie, zmenšování a zvětšování po 10%)

Třetí experiment provádějící převzorkování používá tzv. kaskádu převzorkování. To znamená, že provádí převzorkování již převzorkované fotografie z předchozího kroku. Součástí tohoto postupu je i zmenšování již zmenšené fotografie. Použitím několikanásobného převzorkování vznikají postupně artefakty. Tyto artefakty postupně prováděním jednotlivých iterací experimentu kumulují. Vznik artefaktů je závislý na kvalitě použité metody převzorkování. Změna měřítka není již lineární, ale postupně dochází ke zmenšování zmenšenin. Tento experiment provádí pouze zmenšování původní fotografie. Zmenšuje se postupně o 20%. Osa x popisuje opravdové zmenšení fotografie, nikoliv pouze zmenšení provedené v dané iteraci experimentu. Při provádění postupného zmenšování se vznik artefaktů na výsledku experimentu významně neprojevil. Převzorkovací filtry, interpolační metody a replikace pixelů také zaznamenaly srovnatelnou úspěšnost detekce, vizte obr. 4.2.6.



Obr. 4.2.6: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování o 20%)

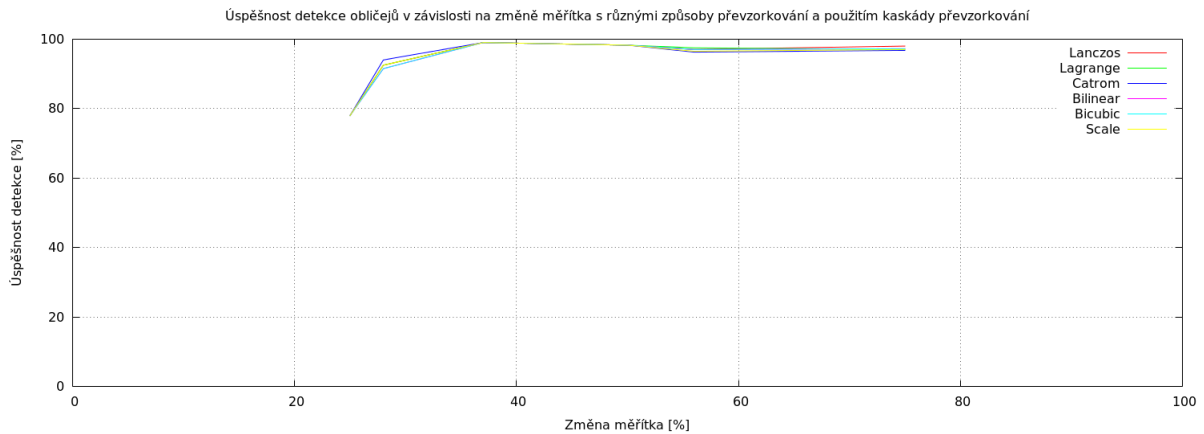
Čtvrtý experiment s převzorkováním je podobný předchozímu experimentu. Nejprve se však provede zmenšení původní fotografie na 25%. Poté se postupně po 20% zvětšuje přibližně do velikosti původní fotografie. Dochází taktéž ke kaskádě převzorkování a zvětšování již zvětšeného obrazu. Zvětšování není lineární. Osa x znázorňuje opravdové zvětšení respektive zmenšení. Při zvětšování překvapivě klesala úspěšnost dvou kvalitních převzorkovacích filtrů, vizte obr. 4.2.7.



Obr. 4.2.7: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšení na 25%, zvětšování o 20%)

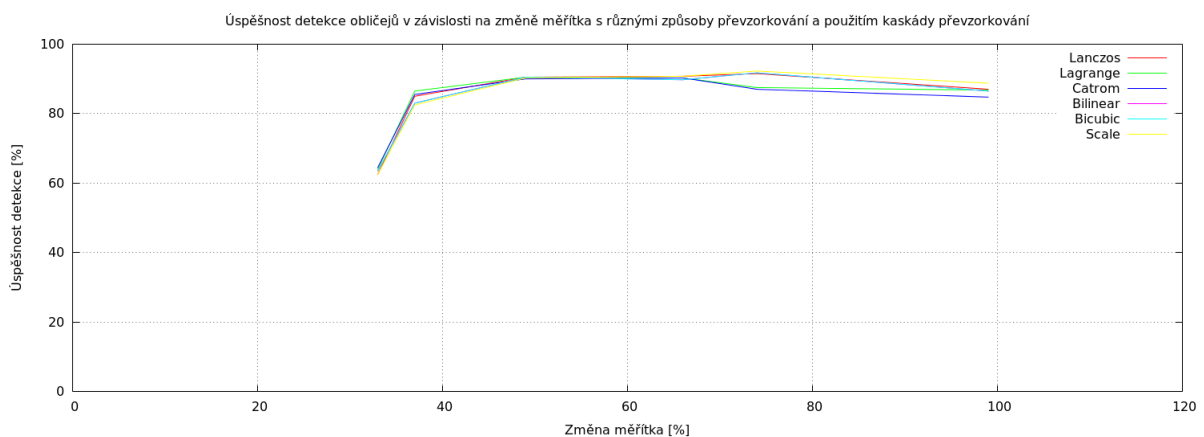
Pátý experiment provádí převzorkování již převzorkované fotografie a zmenšování zmenšenin podobně jako třetí experiment. Rozdíl je v kroku, struktuře kaskády a způsobu převzorkování. Zmenšení se provádí na velikost 3/4 původní fotografie nebo fotografie z předchozího kroku experimentu. Zmenšení přímo na 3/4 se provádí jen u prvních dvou kroků. Poté se zmenšuje o 1/2, která poskytuje přibližně stejné výsledky jako zmenšení na 3/4. Tato hodnota je použita pro

menší zaokrouhlovací chybu a usnadnění výpočtu. Zmenšování o 1/2 nenavazuje přímo na druhý krok experimentu. Třetí krok se provádí z původní fotografie. Čtvrtý krok navazuje na výstup z prvního kroku. Podobně pátý krok navazuje na výstup z kroku druhého. Další úroveň zmenšení navazuje stejným způsobem na kroky 3 až 5. U zmenšení na 3/4 jsou použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se používá dolnoproputsní Gaussův 2D filtr [19] a redukuje se vzorky fotografie ve vertikálním i horizontálním směru [15]. Převzorkovací filtry zaznamenaly srovnatelnou úspěšnost jako interpolační metody a replikace pixelů, vizte obr. 4.2.8.



Obr. 4.2.8: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování na 3/4 fotografie z předchozího kroku)

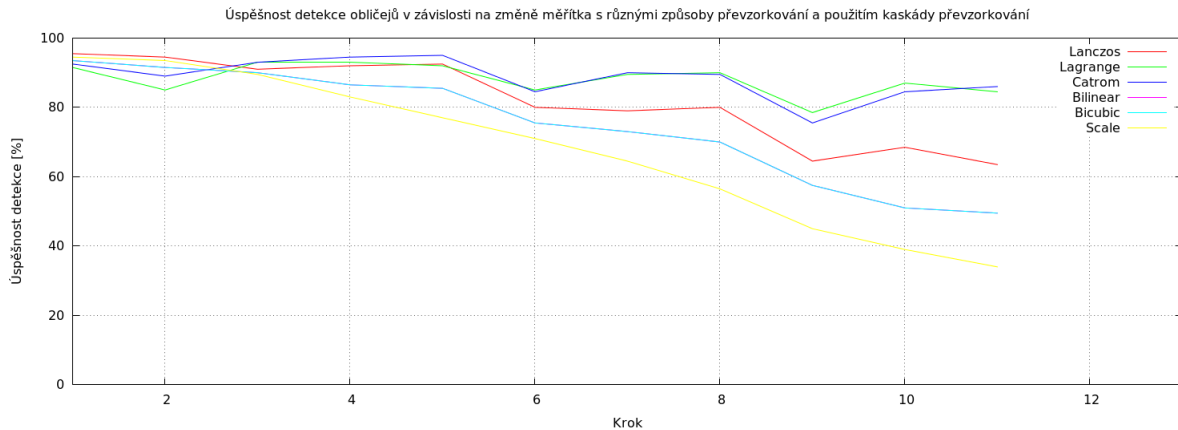
Šestý experiment je velmi podobný předchozímu experimentu. Na rozdíl od předchozího experimentu se však používá předzpracování vstupní fotografie. V rámci tohoto předzpracování se provádí ořez obličejů navíc s 25 pixely na každé straně fotografie. Nejprve se fotografie zvětšuje na 133% velikosti předzpracované fotografie. Následně je postup stejný jako u předchozího experimentu. Zmenšení na 3/4 se provádí u prvních dvou kroků. Poté se zmenšuje o 1/2 jako v předchozím experimentu stejným způsobem. U zmenšení na 3/4 jsou také použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se taktéž používá dolnoproputsní Gaussovo jádro [19] a redukuje se řádky a sloupce pixelů [15]. Převzorkovací filtry zaznamenaly srovnatelnou úspěšnost jako interpolační metody a replikace pixelů, vizte obr. 4.2.9.



Obr. 4.2.9: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez obličejů, zvětšení na 133%, zmenšování na 3/4)

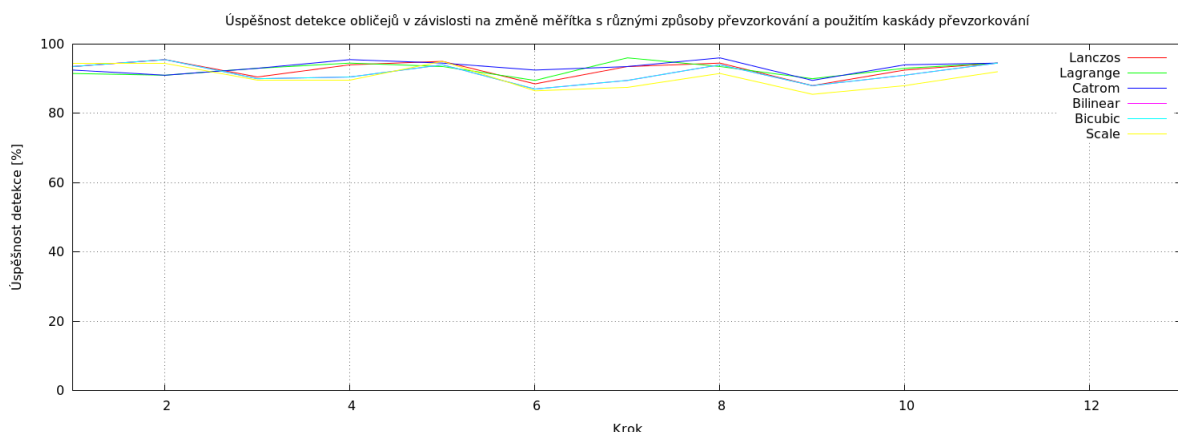
Sedmý experiment je velmi podobný předchozímu experimentu. Na rozdíl od předchozího experimentu se však používá odlišné předzpracování vstupní fotografie. V rámci tohoto předzpracování se provádí ořez obličejů navíc s 10 pixely na každé straně fotografie. Poté se ořezaný obličej zmenšuje na velikost 31x31 pixelů. Zvětšuje se na 133% v každém kroku experimentu.

Následně je postup stejný jako u předchozího experimentu. Zmenšení na 3/4 se provádí u prvních dvou kroků. Poté se zmenšuje o 1/2 jako v předchozím experimentu stejným způsobem. U zmenšení na 3/4 jsou také použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se taktéž používá dolnoproputní Gaussovo jádro [19] a redukují se řádky a sloupce pixelů [15]. Tento experiment používá odlišné nastavení detektoru než ostatní experimenty. U detektoru je vypnuto zmenšování a jsou sníženy limity velikosti fotografie. Při tomto experimentu se již projevila kvalita převzorkovacích filtrů, které zaznamenaly vyšší úspěšnost detekce než interpolační metody a replikace pixelů, vizte obr. 4.2.10.



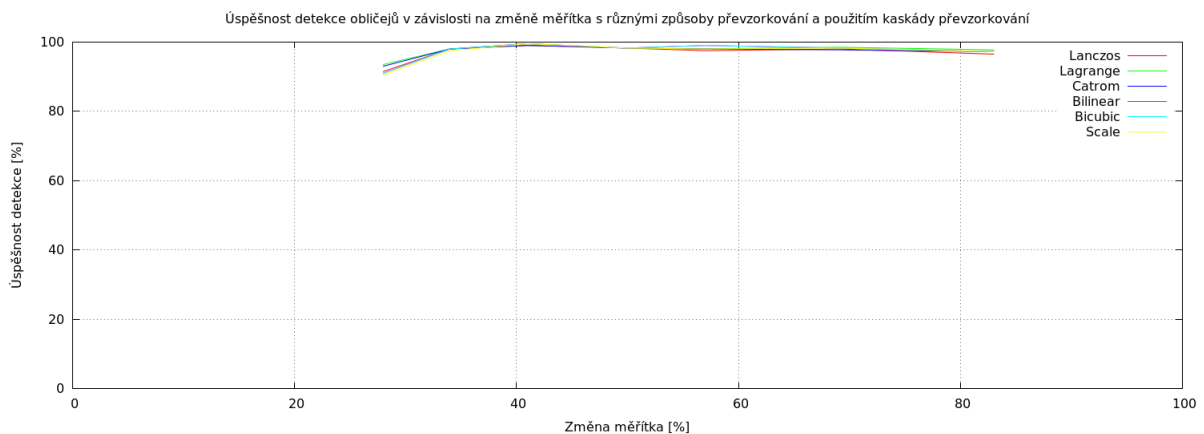
Obr. 4.2.10: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení obličejů, zvětšování na 133% a zmenšování na 3/4 v každém kroku experimentu)

Osmý experiment je velmi podobný předchozímu experimentu. V rámci předzpracování se provádí ořez obličeje navíc s 10 pixely na každé straně fotografie. Poté se ořezaný obličej zmenšuje na velikost 31x31 pixelů. Zvětšuje se na 133% v každém kroku experimentu. Zvětšování se provádí vždy z původní fotografie najednou. Zmenšení na 3/4 se provádí u prvních dvou kroků. Poté se zmenšuje o 1/2 jako v předchozích experimentech. U zmenšení na 3/4 jsou také použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se používá dolnoproputní Gaussovo jádro [19] a redukují se řádky a sloupce pixelů [15]. Tento experiment používá stejné nastavení detektoru jako předchozí experiment. U detektoru je vypnuto zmenšování a jsou sníženy limity velikosti fotografie. Při tomto experimentu se projevila kvalita převzorkovacích filtrů, které zaznamenaly vyšší úspěšnost detekce než interpolační metody a replikace pixelů, vizte obr. 4.2.11.



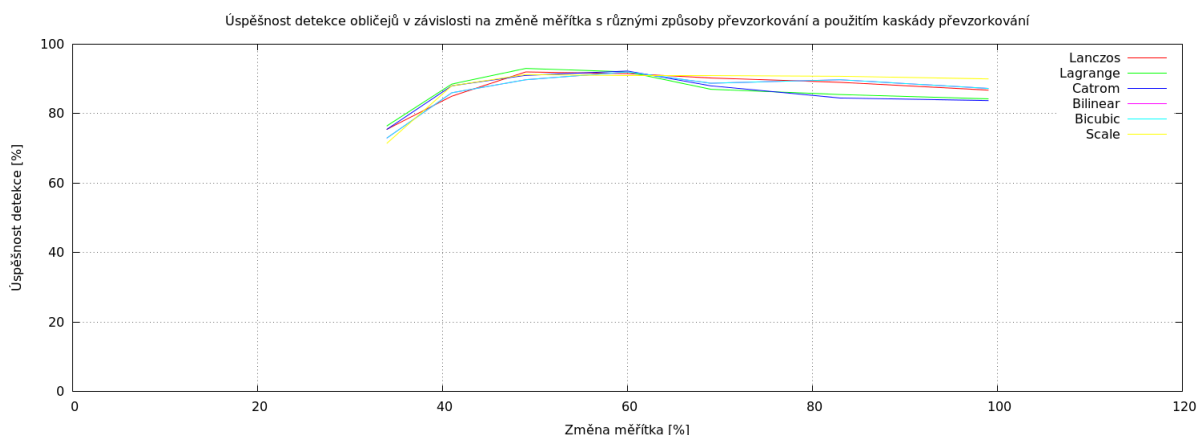
Obr. 4.2.11: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení obličejů, zvětšování na 133% a zmenšování na 3/4 v každém kroku experimentu, zvětšování probíhá najednou)

Devátý experiment je podobný pátému experimentu. Zmenšování se provádí postupně na velikost 5/6 původní fotografie nebo fotografie z předchozího kroku. První tři kroky se provádí přímo zmenšení na 5/6. Poté od čtvrtého kroku se již provádí zmenšení na 1/2 fotografie. Čtvrtý krok však nenavazuje přímo na krok třetí. Ve čtvrtém kroku je použita opět fotografie původní. Pátý krok používá fotografii, která je výstupem z kroku prvního. Šestý krok používá fotografii z druhého kroku a sedmý krok používá fotografii z kroku třetího. Stejným způsobem by mohly v případě potřeby navazovat přímo na kroky 4 až 7 další úrovně zmenšení. Při zmenšování na 5/6 se používají interpolační metody a převzorkovací filtry. U zmenšování na 1/2 se používá dolnopropustní filtr. Jako dolní propust je použito Gaussovo jádro s hodnotou sigma 1.5 [19]. Následně je vymazán každý druhý vzorek fotografie ve vertikálním i horizontálním směru [15]. V tomto experimentu převzorkovací filtry zaznamenaly srovnatelnou úspěšnost jako interpolační metody a replikace pixelů, vizte obr. 4.2.12.



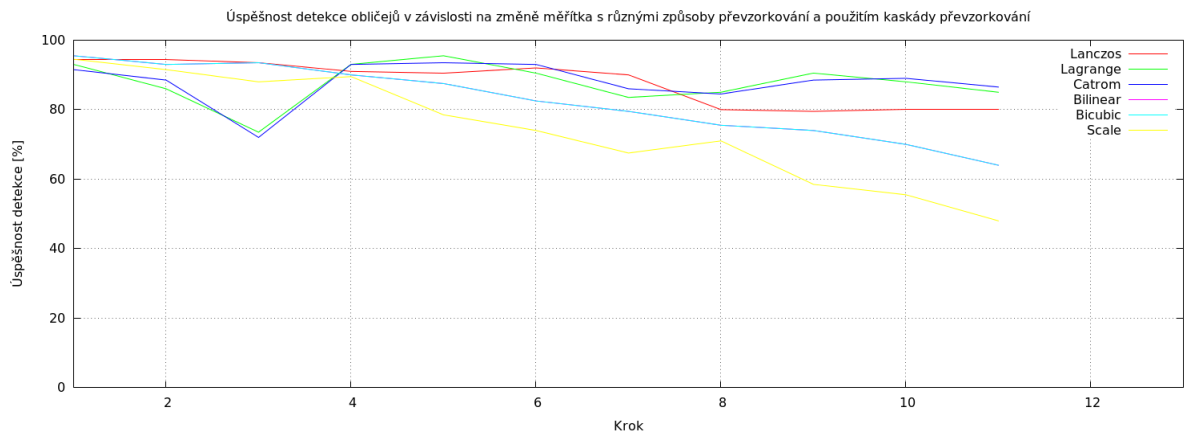
Obr. 4.2.12: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování na 5/6 fotografie z předchozího kroku)

Desátý experiment je velmi podobný předchozímu experimentu. Na rozdíl od předchozího experimentu se používá předzpracování fotografie. V předzpracování se provádí ořez obličejů navíc s 25 pixely na každé straně fotografie. Poté se fotografie zvětšuje na 120%. Následně je postup stejný jako u předchozího experimentu. Zmenšení přímo na 5/6 se provádí u prvních tří kroků. Poté se zmenšuje o 1/2 jako v předchozím experimentu stejným způsobem. U zmenšení na 5/6 jsou také použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se taktéž používá dolnopropustní Gaussovo jádro [19] a redukuje se řádky a sloupce pixelů [15]. Převzorkovací filtry zaznamenaly srovnatelnou úspěšnost jako interpolační metody a replikace pixelů, vizte obr. 4.2.13.



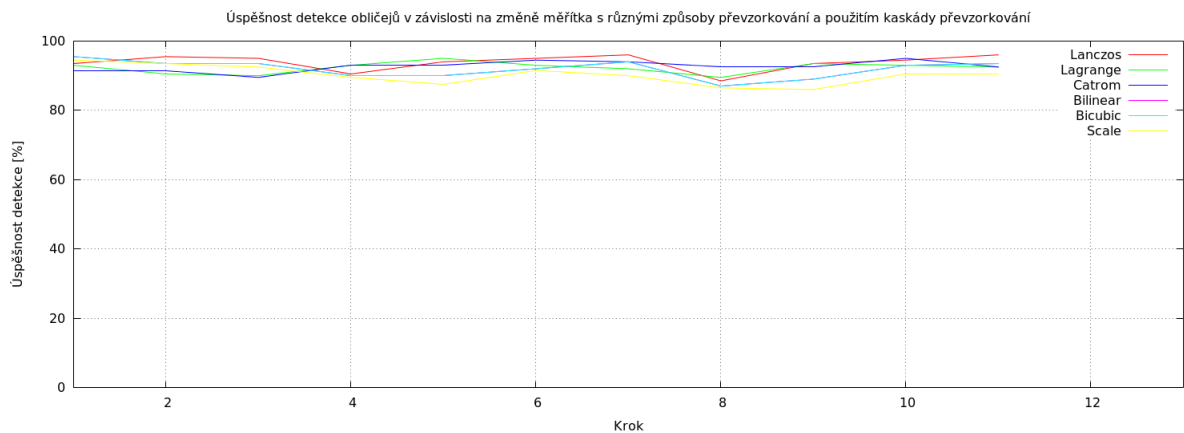
Obr. 4.2.13: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez obličejů, zvětšení na 120%, zmenšování na 5/6 fotografie z předchozího kroku)

Jedenáctý experiment je velmi podobný předchozímu experimentu. Na rozdíl od předchozího experimentu se používá odlišné předzpracování fotografie. V předzpracování se provádí ořez obličeje navíc s 10 pixely a zmenšení ořezaného obličeje na 31x31 pixelů. Fotografie se zvětšuje na 120% v každém kroku experimentu. Následně je postup stejný jako u předchozího experimentu. Zmenšení přímo na 5/6 se provádí u prvních tří kroků. Poté se zmenšuje o 1/2 jako v předchozím experimentu stejným způsobem. U zmenšení na 5/6 jsou také použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se taktéž používá dolnopropustní Gaussovo jádro [19] a redukují se řádky a sloupce pixelů [15]. Tento experiment používá odlišné nastavení detektoru než ostatní experimenty. U detektoru je vypnuto zmenšování a jsou sníženy limity velikosti fotografie. Při tomto experimentu se již projevila kvalita převzorkovacích filtrů, které zaznamenaly vyšší úspěšnost detekce než interpolační metody a replikace pixelů, vizte obr. 4.2.14.



Obr. 4.2.14: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení obličejů, zvětšování na 120% a zmenšování na 5/6 v každém kroku experimentu)

Dvanáctý experiment je podobný předchozímu experimentu. V předzpracování se provádí ořez obličeje navíc s 10 pixely a zmenšení ořezaného obličeje na 31x31 pixelů. Fotografie se zvětšuje na 120% v každém kroku experimentu. Zvětšování probíhá z původní fotografie najednou. Zmenšení přímo na 5/6 se provádí u prvních tří kroků. Poté se zmenšuje o 1/2 jako v předchozím experimentu. U zmenšení na 5/6 jsou použity interpolační metody a převzorkovací filtry. Při zmenšení o 1/2 se používá Gaussovo jádro [19] a redukují se řádky a sloupce pixelů [15]. Tento experiment používá nastavení detektoru jako předchozí experiment. Je vypnuto zmenšování a jsou sníženy limity velikosti fotografie. Při tomto experimentu se projevila kvalita převzorkovacích filtrů, které zaznamenaly vyšší úspěšnost detekce než interpolační metody a replikace pixelů, vizte obr. 4.2.15.



Obr. 4.2.15: Graf úspěšnosti detekce obličejů v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení obličejů, zvětšování na 120% a zmenšování na 5/6 v každém kroku experimentu, zvětšení probíhá najednou)

Poslední třináctý experiment je velmi podobný devátému experimentu. Zmenšování se provádí postupně na velikost 5/6 původní fotografie nebo fotografie z předchozího kroku. První tři kroky se provádí zmenšení přímo na velikost 5/6 fotografie. Poté od čtvrtého kroku se již provádí zmenšení na 1/2 velikosti fotografie. Při zmenšování na 5/6 se používají interpolační metody a převzorkovací filtry. U zmenšování na 1/2 se používá dolnopropustní Gaussovo jádro s hodnotou sigma 1.5 [19]. Následně jsou redukovány řádky a sloupce pixelů [15]. V tomto experimentu se úspěšnost detekce vyhodnocuje odlišným způsobem než u ostatních experimentů. Vyhodnocuje se úspěšnost celé kaskády zmenšování. Výstup tohoto experimentu se nachází v datových souborech.

## 4.3 Vyhodnocení úspěšnosti detekce

Pro automatizované zjištění úrovně úspěšnosti detekce bylo nejprve nutné implementovat porovnávání hodnot souřadnic, výšky a šířky detekovaného objektu s referenčními hodnotami. Dále bylo nutné implementovat vyhodnocení úspěšnosti detekce. Důležité také bylo nastavení určitého rozsahu, kdy je detekce považována ještě za úspěšnou a kdy již nikoliv.

Výstup detektoru je struktura v serializačním formátu JSON. Bylo by nevhodné jej dále převádět, protože formát JSON je pro tuhle činnost přesně odpovídající. Je důležité ještě dodat, že jedna struktura reprezentuje jednu fotografii. Tato struktura obsahuje kromě informací o detekovaném souboru také informace o detekovaných objektech. Mezi informace o detekovaných objektech patří souřadnice x, souřadnice y, výška h a šířka w detekovaného objektu.

Nejefektivnější bylo zvolit ukládání referenčních JSON struktur do externích souborů. Z těchto souborů se poté struktury postupně načítají a zpracovávají. Struktury detekovaných objektů se taktéž ukládají do souborů pro přehlednost a efektivitu zdrojového kódu programu. Program by mohl tyto hodnoty zpracovávat rovnou v rámci běhu a porovnávat s načtenými a zpracovanými referenčními strukturami. Tohle řešení by však bylo nevhodné z hlediska kvality zdrojového kódu. Operace načtení obsahu souboru není příliš časově náročná. Jistě lze namítnout, že pokud bude dataset čítat tisíce fotografií, tak se to časově znatelně projeví. V poměru s časem běhu experimentu je to však zanedbatelné. Tudíž se postupně ukládají, načítají a zpracovávají jak referenční struktury, tak i struktury detekovaných objektů.

Pro účely vyhodnocení úrovně úspěšnosti detekce objektů v obraze byly nejprve vytvořeny tzv. referenční struktury, které byly vytvořeny již předem nezávisle na experimentech. Při tvorbě těchto referenčních JSON struktur nebylo použito žádné předzpracování původní fotografie. Bylo použito základní nastavení detektoru s krokem posunu detekčního okna 1 pixel. Všechny tyto struktury byly postupně jedna po druhé zkontrolovány, zda jsou správné. Struktury byly porovnávány podle počtu detekovaných objektů a jejich souřadnic, výšky a šířky. Součástí tohoto porovnávání při spuštění detektoru bylo použití parametru show, který slouží k zobrazení výsledné detekce přímo na fotografii. Správnost těchto struktur je tedy zaručena. V případě nalezení chyby docházelo k jejímu okamžitému opravení.

V současném stavu jsou referenční JSON struktury uloženy v souborech pojmenovaných podle jednotlivých vstupních fotografií s koncovkou .out. Tyto soubory jsou připraveny pro načtení programem experimentu k porovnávání se strukturami detekovaných objektů ze vstupní fotografie. Prvním krokem programu je otevření adresáře se soubory s referenčními strukturami a načtení jmen všech souborů s koncovkou .out.

```
1 [  
2 {"file": "/home/beda/Plocha/LBPDetector/bands_120_1/  
   img_001.jpg", "nf": 3981231, "ns": 879960, "objects": [[206, 60, 60, 60, 40]]}  
3 ]
```

Obr. 4.4.1: Referenční JSON struktura

Stejný postup je zapotřebí také pro struktury uložené z detekce programu experimentu. Poté se prochází jednotlivé soubory, ve kterých jsou uloženy JSON struktury. Pomocí filtračního programu jq, který snadno vybere požadovaná data z JSON struktury, se získávají objekty. Z vyfiltrovaných

objektů se získají opět pomocí programu jq souřadnice, výška a šířka objektu. Tento postup je použit jak pro referenční struktury, tak pro výstupní struktury z detekce objektů.

Dále byla zvolena požadovaná rezerva respektive rozsah, kdy je ještě detekce považována za úspěšnou. Tento rozsah byl určen pro souřadnice, šířku a výšku objektu. Určený rozsah se poté přičte a odečte od referenčních hodnot souřadnic, šířky a výšky objektu. Poté se hodnoty  $x$ ,  $y$ ,  $w$ ,  $h$  ze struktury detekovaného objektu porovnají s hodnotami referenčních struktur rozšířenými o rezervy neboli rozsahy. Pokud hodnoty detekovaného objektu vycházejí do těchto rozsahů, lze považovat detekci objektu v obraze za úspěšnou. Pokud ne, jedná se o neúspěch, který se musí také ve výsledcích experimentů projevit.

Rezervy neboli rozsahy úspěšnosti jsou pro jednotlivé experimenty nastaveny různě. Použito je přesněji řečeno dvojí nastavení. Toto nastavení závisí na konkrétním předzpracování vstupní fotografie před detekcí objektů. Pokud nedochází v experimentu ke změně měřítka vstupní fotografie v rámci předzpracování, je rozsah úspěšnosti nastaven tzv. napevno. To znamená, že se tato hodnota v průběhu experimentu již nemění. Konkrétní hodnota rozsahu úspěšnosti je pro daný experiment zvolena na základě provedených pokusů. Pokud dochází v rámci předzpracování ke změně měřítka vstupní fotografie, určení rozsahu úspěšnosti se provádí odlišným způsobem. Vychází se také z referenčních hodnot, ale jsou přepočítány dle aktuálního zvětšení nebo zmenšení fotografie. Přesný postup výpočtu je uveden v podkapitole vzorkování.

Výpočet celkové úspěšnosti detekce celé fotografie se vypočítá pomocí jednoduché rovnice. Pro její výpočet je nutné vést si informaci o počtu úspěšně a také neúspěšně detekovaných objektů. Nejprve se vynásobí počet úspěšně detekovaných objektů 100. Získaná hodnota se poté vydělí součtem počtu úspěšných a neúspěšných detekcí objektů. Výsledkem je procentuální úspěšnost detekce objektů v obraze dané fotografie. Tyto výsledky se v rámci daného experimentu ukládají do datového souboru, který tomuto experimentu přísluší. Tento datový soubor shromažďuje data z daného experimentu. To znamená, že pro každý experiment je výstupem jeden nebo i více takových datových souborů. Tyto datové soubory mohou sloužit uživateli jako zdroj tabulkových informací. Z těchto datových souborů se nakonec použijí data pro vygenerování grafů. Tyto grafy slouží pro přehledné zobrazení výsledků experimentů. Pomocí těchto grafů lze snadno nalézt optimální parametry pro nastavení detektoru obličejů a optimální předzpracování fotografie.

```
1 # Data file for Gnuplot
2 # VUT FIT v Brne
3 # Cernosek Bedrich
4 # xcerno19@stud.fit.vutbr.cz
5
6 File      Scale   Success
7 img_0    10      0
8 img_0    100     100
9 img_0    110     100
10 img_0   120     100
11 img_0   130     100
12 img_0   140     100
13 img_0   150     100
14 img_0   160     100
15 img_0   170     100
16 img_0   180     100
17 img_0   190     100
18 img_0   20      0
19 img_0   200     100
20 img_0   210     100
```

Obr. 4.4.2: Datový soubor obsahující výstupní data experimentu

Vyhodnocení úspěšnosti detekce objektů všech parametrů experimentu na dané fotografii funguje obdobně jako vyhodnocení pro jeden parametr experimentu. Prochází se struktury detekovaných objektů pro všechny parametry experimentu, které náleží dané fotografii. Každá struktura, která je výstupem detektoru se porovná s referenční strukturou. Porovnání je stejné jako v případě předchozího vyhodnocení. Výsledek porovnání se ukládá do datové struktury. Pro tento účel



je použito jednodimenzionální pole. V případě neúspěchu se ukládá nula. Nula však nesmí přepsat jedničku v případě předchozího úspěchu. Pokud je detekce úspěšná, ukládá se do pole na daný index jednička. Po provedení vyhodnocení struktur všech parametrů experimentu se provádí celkové vyhodnocení úspěšnosti dané fotografie. Vypočítá se z pole počet úspěšných a neúspěšných detekcí. Počet úspěšných detekcí se vynásobí 100. Výsledek se vydělí součtem úspěšných a neúspěšných detekcí. Výsledkem je procentuální úspěšnost detekce na dané fotografii v rámci všech parametrů experimentu. Výsledek po provedení experimentu je uložen v datových souborech.

## 4.4 Zobrazení výsledků experimentů

K zobrazení výsledků experimentů byl použit program Gnuplot. Tento program umožňuje jednoduše a velmi kvalitně vykreslovat libovolné grafy. Výsledné grafy se vynášejí z datových souborů, které obsahují data z příslušných experimentů. Graf se pomocí tohoto programu vykresluje jednoduše použitím příkazu `gnuplot`. Jako parametr při spouštění tohoto programu se používá konfigurační soubor s koncovkou `.gpl`. Veškeré nastavení programu `gnuplot` se provádí pomocí příkazu `set`, vizte obr. 4.5. Nejprve je nutné nastavit tzv. terminál, který určuje formu výstupu programu, respektive formát výsledného grafu. Pro nastavení výstupního formátu se používá terminál `pngcairo`, což je kvalitnější verze terminálu `png`. Terminál se nastavuje pomocí příkazu `set terminal`. Současně se nastavuje pomocí příkazu `size` velikost výsledného grafu. Tato velikost se nastavuje pevně na 960x530 pixelů pro menší grafy a pro větší grafy 1500x550 pixelů. Pro nastavení výstupního souboru slouží příkaz `set output`, jehož argumentem je název výstupního souboru, do kterého bude graf uložen. Nastavit titulek grafu lze použitím příkazu `set title`, za který se již uvádí titulek v uvozovkách. Pro nastavení rozsahu os grafu slouží příkaz `set xrange` a `set yrange`. Rozsah se uvádí do hranatých závorek, ve kterých je počáteční a konečná hodnota oddělena dvojtečkou. Mřížku grafu lze zobrazit použitím příkazu `set grid x y`. Výsledný graf se vygeneruje pomocí příkazu `plot`, jehož argumentem je datový soubor s koncovkou `.dat`. Dále je možno blíže specifikovat výběr dat z datového souboru příkazem `using`, jež určí sloupce dat, které budou k vykreslení použity. Poté následuje titulek zadaný již známým příkazem `title`, který popisuje vykreslenou funkci v grafu. Pro proložení vykreslovaných bodů přímkou lze použít příkaz `smooth`.

```
1 # Gnuplot script file for plotting data
2 # VUT FIT v Brne
3 # Cernosek Bedrich
4 # xcerno19@stud.fit.vutbr.cz
5
6 set terminal pngcairo size 1500,550
7 set output "resample_ext_lin.png"
8
9 set title "Úspěšnost detekce obličejů v závislosti na změně měřítka
   s různými způsoby převzorkování"
10 set xlabel "Změna měřítka [%]"
11 set ylabel "Úspěšnost detekce [%]"
12
13 set xrange [10:350]
14 set yrange [0:100]
15 set grid x y
16
17 plot "lanczos.dat" using 2:3 title 'Lanczos' smooth unique,
   "lagrange.dat" using 2:3 title 'Lagrange' smooth unique,
   "catrom.dat" using 2:3 title 'Catrom' smooth unique, "bilinear.dat"
   using 2:3 title 'Bilinear' smooth unique, "bicubic.dat" using 2:3
   title 'Bicubic' smooth unique, "scale.dat" using 2:3 title 'Scale'
   smooth unique
```

Obr. 4.5: Konfigurační soubor programu Gnuplot pro tvorbu grafů

## 4.5 Tvorba datasetu

Testovací dataset tvoří pouze pozitivní snímky. Pozitivními snímky se rozumí fotografie, na kterých se nachází minimálně jeden objekt na každé fotografii. Přesněji řečeno, na každé z fotografií datového setu se nachází právě jeden obličej. Algoritmus experimentu je však schopen při detekci správně porovnat i více detekovaných objektů. Výpočet úspěšnosti detekce objektů také počítá s možností výskytu více objektů na jedné fotografii. Výpočet procentuální úspěšnosti je této situaci také přizpůsoben. Testovací dataset byl takto zvolen z důvodu věrohodnosti, porovnatelnosti a posouzení výsledků detekce obličejů v obraze. S rostoucím počtem obličejů na fotografii úspěšnost detekce za určitých podmínek výrazně klesá. Mezi tyto podmínky patří různá velikost obličejů, respektive jejich výrazný rozdíl. Dále má na úspěšnost detekce velký vliv příliš mnoho obličejů ve vzájemné blízkosti, například při tzv. skupinových fotografiích a fotografiích ze společenských akcí. Konkrétním příkladem může být fotografie divadelního publika, fotografie z hudebního koncertu a podobně. Detekci může taktéž negativně ovlivňovat tmavé prostředí, ve kterém by mohla být fotografie pořízena. Vliv na úspěšnost detekce může mít i barva kůže, sluneční brýle, doplňky a zdobnosti fotografované osoby. Tato zjištění se opírají o mé pokusy s detektorem.

Dalšími problémy, se kterými se může detektor obličejů v obraze setkat, jsou různé stíny, odrazy světla nebo mračna. Tyto objekty může detektor označit za regulární obličej. Díky referenčním strukturám se však snadno odhalí, že se o pravé lidské obličej nejedná. Tato skutečnost se projevuje snížením výsledné úspěšnosti detekce na dané fotografii. Dalším problémem mohou být obličej na obrazech, různé portréty, fotografie ve fotografii, potisk obličej na oblečení a podobně. Takové fotografie, které by mátl detektor byly z datasetu vyřazeny. Tyto falešné obličej jsou ve své podstatě správně detekované, ale nelze je považovat za opravdové obličej. Nepravé obličej je obtížné v rámci detektoru odhalit.

Výsledný dataset, který vznikl pro ověření funkčnosti algoritmu vyhodnocení úspěšnosti detekce a také pro experimenty čítá 120 fotografií. Tyto fotografie jsou všechny kvalitní. Jsou pořízeny fotografy a často vylepšeny a upraveny ve fotografických nástrojích. Vyfotografované osoby na fotografiích datasetu jsou součástí různých subkultur. U těchto osob se často vyskytuje tetování v oblasti tváře, piercingy, plugy, naušnice, extravagantní účesy zakrývající části obličej a podobně. Tyto skutečnosti však neměly významný vliv na úspěšnost detekce obličejů. Detektor obličejů si s nimi velmi dobře poradil. V datasetu se však nenachází obličej se slunečními brýlemi. Sluneční brýle již občas detektoru působily problémy. Pozadí na fotografiích je velmi různorodé.

## 4.6 Zhodnocení výsledků

Výsledek experimentu s krokem posunu detekčního okna prokázal, že je vhodné pro detekci obličejů v obraze použít krok posunu 2 pixely. Úspěšnost detekce s tímto nastavením se pohybovala přibližně kolem 95%, což je skvělý výsledek. S použitím kroku posunu okna 3 pixely již úspěšnost mírně klesla na 87%. Tento výsledek je ovšem taktéž velmi příznivý. Krok posunu detekčního okna 4 pixely zaznamenal také mírný propad úspěšnosti. Toto nastavení má úspěšnost na testovaném datasetu 80%. Podle mého názoru je tahle hodnota na hranici použitelnosti pro nastavení detektoru. Experiment s rotací detekovaného objektu zaznamenal očekávaně největší úspěšnost při nulové rotaci. Rotace od  $-20^\circ$  do  $25^\circ$  poskytovaly přijatelné výsledky s úspěšností nad 80%. Menší a větší hodnoty rotací pro předzpracování nemají podle mého názoru význam. Úspěšnost u těchto hodnot znatelně s každým krokem experimentu klesá. Zajímavostí je asymetričnost kladné a záporné rotace, která je způsobena pravděpodobně různým natočením obličejů. Pravděpodobně tohle drobné natočení je v jednom směru znatelnější než v druhém. Asymetričnost je tedy dána charakteristikou použitého datasetu. Zajímavé výsledky poskytl experiment se zkosením pomocí afinní transformace. Hodnoty parametrů od  $-0.4$  do  $0.3$  u osy  $x$  dosahují úspěšnosti nad 80%. U osy  $y$  hodnoty parametrů od  $-0.4$  do  $0.4$  dosahují také nad 80% úspěšnosti. Experiment se zkosením poskytoval také asymetrické výsledky u obou transformací. Stejně jako u rotace to bylo dáno vlastnostmi použitého datasetu. Převzorkovací filtry zaznamenaly vyšší úspěšnost než interpolační metody a replikace pixelů u lineárních

experimentů přibližně o 5 až 10%. Při použití tzv. kaskády převzorkování docházelo ke vzniku artefaktů. Tato kaskáda znamená použití v každém kroku experimentu již převzorkovaný obraz z předchozí iterace. Postupně tedy docházelo ke kumulaci artefaktů. Při použití kaskády převzorkování jsou již patrnější rozdíly mezi výsledky převzorkovacích filtrů, interpolačních metod a replikace pixelů. Lagrangeův filtr a Catmull-Rom filtr zaznamenaly přibližně 90% úspěšnost. Lanczosův filtr měl úspěšnost 80%. Interpolační metody zaznamenaly přibližně 65% úspěšnost detekce. Změna měřítka fotografie s použitím replikace pixelů měla úspěšnost 50%. Tyto výsledky byly získány opakovaným převzorkováním ořezaných a zmenšených obličejů. Při tomto experimentu bylo vypnuto zmenšování u detektoru. Bylo provedeno 11 kroků experimentu. V každém kroku se převzorkování provádělo u zvětšování i zmenšování.

Vyhodnocení úspěšnosti detekce objektů funguje správně a efektivně. Vytvořený algoritmus vyhodnocení úspěšnosti bohužel nebylo možné porovnat s jiným řešením, protože žádné řešení tohoto typu neexistuje nebo není rozšířeno. Rozsahy úspěšnosti detekce jsou vhodně nastaveny pro použitý dataset. Je tedy možné, že pro použití s jinými datasey nebudou příliš vhodné. Nastavení hodnot rozsahů úspěšnosti je však velmi jednoduché a nezabere příliš mnoho času. Pro vyhodnocení úspěšnosti, jak bylo konkrétně zmíněno v práci, jsou nutné tzv. referenční struktury. Tyto struktury jsou vázány k danému datasetu. Tudíž referenční struktury k použitému datasetu jsou pro jiný dataset zcela nepoužitelné. Získané výstupy z automatizovaného experimentu byly manuálně porovnány spuštěním detektoru z příkazového řádku. Fotografie byly v požadovaném předzpracování a s požadovanými parametry detektoru. Je tedy zaručena správnost algoritmu vyhodnocení úspěšnosti.

## 5 Závěr

Cílem této bakalářské práce bylo vytvořit algoritmus pro vyhodnocení úspěšnosti detekce objektů v obraze. Bylo za úkol vytvořit dataset, na kterém byla ověřena správnost vyhodnocení úspěšnosti detekce. Zadáno bylo také vytvoření experimentů, z jejichž výsledků se získaly optimální parametry pro detektor obličejů a optimální předzpracování fotografií před detekcí obličejů.

Práce na výše uvedených úkolech probíhala podle plánu. Podařilo se úspěšně splnit všechny úkoly dané zadáním práce. Byl vytvořen algoritmus pro vyhodnocení úspěšnosti detekce objektů v obraze. Tento algoritmus porovnává referenční hodnoty s hodnotami detekovaných objektů. Mezi tyto hodnoty patří souřadnice a velikost detekovaného objektu. Po porovnání hodnot algoritmus vypočítá úspěšnost detekce objektů v obraze a procentuálně ji vyjádří. Vyhodnocení úspěšnosti detekce objektů na dané fotografii lze provést v rámci daného parametru experimentu nebo v rozsahu celého experimentu. Byl vytvořen dataset, na kterém byla správnost algoritmu vyhodnocení úspěšně ověřena. Proběhla taktéž manuální kontrola, při které se výstup algoritmu postupně porovnával s výstupem detektoru. Dále byly vytvořeny experimenty s nastavením detektoru a předzpracováním fotografie. Tyto experimenty taktéž otestovaly správnost algoritmu vyhodnocení úspěšnosti detekce objektů. Zároveň jejich výstupem bylo optimální nastavení detektoru a optimální předzpracování vstupních fotografií.

Provedením experimentů na použitém datasetu se podařilo získat následující výsledky. Krok posunu detekčního okna do 4 pixelů má úspěšnost nad 80%. Rotace od  $-20$  do  $25^\circ$  má úspěšnost nad 80%. Zkosení pomocí afinní transformace od  $-0.4$  do  $0.3$  u osy  $x$  dosahuje úspěšnosti nad 80%, od  $-0.4$  do  $0.4$  u osy  $y$  dosahuje také nad 80% úspěšnosti. Změna měřítka od 25% do 200% s různými způsoby převzorkování má úspěšnost detekce u lineárních experimentů nad 80%. Převzorkovací filtry zaznamenaly u těchto experimentů o 5 až 10% vyšší úspěšnost než interpolační metody a replikace pixelů. Při použití kaskády převzorkování se již více projevila kvalita převzorkovacích filtrů na úspěšnosti detekce. Nejlepší výsledky při opakovaném převzorkování zaznamenal Lagrangeův filtr a Catmull-Rom filtr, jejichž úspěšnost byla přibližně 90%. Lanczosův filtr zaznamenal úspěšnost 80%. Úspěšnost interpolačních metod byla přibližně 65%. Změna měřítka s použitím replikace pixelů měla 50% úspěšnost. Tyto výsledky byly získány po 11 krocích experimentu, ve kterých se převzorkování provádělo u zvětšení i zmenšení fotografie.

Díky této práci jsem pochopil, jak detekce objektů v obraze funguje. Poznal jsem více přístupů k detekci objektů a naučil jsem se algoritmy, které se používají k její implementaci. Práce mi dala celkový rozhled v oblasti detekce objektů v obraze. Prohloubil jsem si znalosti z oblasti zpracování obrazu, zpracování signálů a matematické analýzy.

V této bakalářské práci je možné pokračovat a nadále ji vylepšovat. Pro představu uvádím níže několik nápadů pro další možnou práci. Vhodné by bylo provést různé kombinace experimentů. Zajímavé by také bylo zobrazit tyto kombinace. Výsledkem tohoto zobrazení by byl čtyř nebo i vícedimenzionální graf. Zajímavý výstup experimentu by mohlo poskytnout provádění zkosení pomocí afinní transformace v obou dimenzích zároveň. Určitě zajímavé by bylo zobrazit graf experimentu se zkosením pomocí afinní transformace ve třech dimenzích. V rámci předzpracování by bylo zajímavé provádět stlačení (snížení a zúžení) fotografie postupně v obou dimenzích. Vhodné by bylo rozšíření datasetu pro získání většího množství dat a tím kvalitnějšího výsledku. U rotace a afinní transformace by bylo vhodné navrhnout a implementovat inteligentnější způsob výpočtu rozsahů úspěšnosti.

Algoritmus vyhodnocení úspěšnosti detekce je vhodně navržen, lze tudíž použít pro detekci libovolných objektů. Je možné použít libovolný detektor objektů. Pro předzpracování lze použít libovolný nástroj. Detektor nebo nástroj může být jednoduše spouštěn v rámci skriptu stejně jako v příkazovém řádku terminálu. Lze také využít libovolného programovacího jazyku a výsledný program spouštět ve vytvořeném skriptu. Je možné také použít libovolný dataset pro vyhodnocení úspěšnosti. Samozřejmostí je libovolné nastavení parametrů experimentu. Možnosti jsou velmi široké a znamenají pouze minimální zásah do zdrojového kódu skriptu.

# Literatura

- [1] FREUND, Y., SCHAPIRE, R.: A short introduction to boosting. Society for Artificial Intelligence, 14(5), p. 771-870, (1999).  
Dostupné z WWW: <http://cseweb.ucsd.edu/~yfreund/papers/IntroToBoosting.pdf>
- [2] SCHAPIRE, Robert: Explaining AdaBoost. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.cs.princeton.edu/~schapire/papers/explaining-adaboost.pdf>
- [3] VIOLA, P., JONES, M.: Robust Real-time Object Detection. Int. J. Comput. Vision 57(2), (May 2004). Dostupné z WWW:  
<https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-IJCV-01.pdf>
- [4] KADLČEK, F., FUČÍK, O.: Evolutionary design of Local Binary Pattern feature shapes for object detection. NASA/ESA Conference on Adaptive Hardware Systems, Nuremberg, Germany, s. 8, (2012). Dostupné z WWW:  
<http://www.fit.vutbr.cz/~fucik/pubs.php?file=%2Fpub%2F9987%2FPID2320815.pdf&id=9987>
- [5] JURÁNEK, R., ZEMČÍK, P., HRADIŠ, M.: Real-Time Algorithms of Object Detection Using Classifiers. Real-Time Systems, Rijeka:InTech, s. 1-22, ISBN 9789535105107, (2012).  
Dostupné z WWW: <http://cdn.intechopen.com/pdfs-wm/35324.pdf>
- [6] ŠOCHMAN, J., MATAS, J.: WaldBoost Learning for Time Constrained Sequential Detection. CVPR IEEE Computer Society Conference, San Diego, CA, USA, p. 150-156, Vol. 2, ISBN 0-7695-2372-2 (2005).  
Dostupné z WWW: <http://cmp.felk.cvut.cz/~matas/papers/sochman-waldboost-cvpr05.pdf>
- [7] DUCHON, Claude E.: Lanczos Filtering in One and Two Dimensions. Journal of Applied Meteorology. American Meteorological Society. Volume 18. (August 1979).  
Dostupné z: [http://www.unl.edu.ar/ceneha/uploads/LanczosFiltering\(1979\).pdf](http://www.unl.edu.ar/ceneha/uploads/LanczosFiltering(1979).pdf)
- [8] KHALAJZADEH, H., MANSOURI, M., TESHNEHLAB, M.: Face Recognition using Convolutional Neural Network and Simple Logistic Classifier. WSC17, 2012 Online Conference on Soft Computing in Industrial Application. (December 2012). Dostupné z: [http://www.dap.vsb.cz/wsc17conf/Media/Default/Page/online\\_wsc17\\_submission\\_59.pdf](http://www.dap.vsb.cz/wsc17conf/Media/Default/Page/online_wsc17_submission_59.pdf)
- [9] LAWRENCE, S., GILES, L., TSOI, A. H., BACK, A. D.: Face Recognition: A convolutional Neural Network Approach. IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition.  
Dostupné z: <http://www.clgiles.ist.psu.edu/papers/IEEE.TNN.face.recognition.hybrid.nn.pdf>
- [10] FASEL, B: Robust Face Analysis using Convolutional Neural Networks. IEEE. (2002). [online]. Dostupné z: [http://www.hci.iwr.uni-heidelberg.de/publications/dip/2002/ICPR2002/DATA/04\\_2\\_4.PDF](http://www.hci.iwr.uni-heidelberg.de/publications/dip/2002/ICPR2002/DATA/04_2_4.PDF)
- [11] SMITH, K., ROSS, A.: Face Recognition. National Science and Technology Council, Committee on Technology, Committee on Homeland and National Security, Subcommittee on Biometrics. (August 2006). [online]  
Dostupné z: <http://www.biometrics.gov/Documents/FaceRec.pdf>

- [12] TURK, M., PENTLAND, A.: Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*. Vol. 3, No. 1, p. 71-86. (1991).  
Dostupné z: <http://www.cs.ucsb.edu/~mturk/Papers/jcn.pdf>
- [13] TURK, M., PENTLAND, A.: Face Recognition Using Eigenfaces. [online].  
Dostupné z: <http://www.cs.ucsb.edu/~mturk/Papers/mturk-CVPR91.pdf>
- [14] THYSSEN, Anthony. ImageMagick: Cutting and Bordering. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.imagemagick.org/Usage/crop/>
- [15] THYSSEN, Anthony. ImageMagick: Resize or Scaling. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.imagemagick.org/Usage/resize/>
- [16] THYSSEN, Anthony. ImageMagick: Affine Matrix Transforms. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.imagemagick.org/Usage/distorts/affine/>
- [17] THYSSEN, Anthony. ImageMagick: Resampling Filters. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.imagemagick.org/Usage/filter/>
- [18] THYSSEN, Anthony. ImageMagick: Miscellaneous. [online]. [cit. 2015-03-10].  
Dostupné z: <http://www.imagemagick.org/Usage/misc/>
- [19] THYSSEN, Anthony. ImageMagick: Convolution of Image. [online]. [cit. 2015-05-01].  
Dostupné z: <http://www.imagemagick.org/Usage/convolve/>
- [20] THYSSEN, Anthony. ImageMagick: Warping. [online]. [cit. 2015-05-01].  
Dostupné z: <http://www.imagemagick.org/Usage/warping/>

# Seznam obrázků

- Obr. 4.2.1: Graf úspěšnosti detekce v závislosti na velikosti kroku posunu detekčního okna
- Obr. 4.2.2: Graf úspěšnosti detekce v závislosti na rotaci
- Obr. 4.2.3: Graf úspěšnosti detekce v závislosti na afinní transformaci
- Obr. 4.2.4: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (původní fotografie, pravidelný krok)
- Obr. 4.2.5: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (původní fotografie, ořez, pravidelný krok)
- Obr. 4.2.6: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování o 20%)
- Obr. 4.2.7: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšení na 25%, zvětšování o 20%)
- Obr. 4.2.8: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování na 3/4)
- Obr. 4.2.9: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez, zvětšení na 133%, zmenšování na 3/4)
- Obr. 4.2.10: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení, zvětšování na 133%, zmenšování na 3/4)
- Obr. 4.2.11: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení, zvětšování na 133%, zmenšování na 3/4, zvětšování najednou)
- Obr. 4.2.12: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, zmenšování na 5/6)
- Obr. 4.2.13: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez, zvětšení na 120%, zmenšování na 5/6)
- Obr. 4.2.14: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení, zvětšování na 120%, zmenšování na 5/6)
- Obr. 4.2.15: Graf úspěšnosti detekce v závislosti na změně měřítka s různými způsoby převzorkování (kaskáda převzorkování, ořez a zmenšení, zvětšování na 120%, zmenšování na 5/6, zvětšování najednou)
- Obr. 4.4.1: Referenční JSON struktura
- Obr. 4.4.2: Datový soubor obsahující výstup experimentu
- Obr. 4.5: Konfigurační souboru programu Gnuplot pro tvorbu grafů

# Seznam příloh

## Příloha 1. CD

Zdrojové texty experimentů:

```
det_exp_cas_34.sh
det_exp_cas_34_b.sh
det_exp_cas_34_c.sh
det_exp_cas_34_d.sh
det_exp_cas_56.sh
det_exp_cas_56_b.sh
det_exp_cas_56_c.sh
det_exp_cas_56_d.sh
det_exp_cas_56_all.sh
det_exp_cas_min.sh
det_exp_cas_max.sh
det_exp_ext_lin.sh
det_exp_ext_lin_b.sh
det_exp_str_rot_aff.sh
```

Pomocné skripty:

```
txt_to_json.sh
ref_make.sh
```

Referenční struktury (anotace):

```
exp_ref/img_001.out, ...
exp_ref_all/img_001.out, ...
faces_ref/img_0.out, ...
```

Datové soubory:

```
affine.dat
affine2.dat
bicubic.dat
bilinear.dat
catrom.dat
lagrange.dat
lanczos.dat
rotation.dat
scale.dat
stride.dat
```

Konfigurační soubory pro tvorbu grafů:

```
affine.gpl
resample_cas_34.gpl
resample_cas_34_b.gpl
resample_cas_34_c.gpl
resample_cas_34_d.gpl
resample_cas_56.gpl
resample_cas_56_b.gpl
resample_cas_56_c.gpl
resample_cas_56_d.gpl
```



resample\_cas\_min.gpl  
resample\_cas\_max.gpl  
resample\_ext\_lin.gpl  
resample\_ext\_lin\_b.gpl  
rotation.gpl  
stride.gpl

Grafy experimentů:

affine.png  
resample\_cas\_34.png  
resample\_cas\_34\_b.png  
resample\_cas\_34\_c.png  
resample\_cas\_34\_d.png  
resample\_cas\_56.png  
resample\_cas\_56\_b.png  
resample\_cas\_56\_c.png  
resample\_cas\_56\_d.png  
resample\_cas\_min.png  
resample\_cas\_max.png  
resample\_ext\_lin.png  
resample\_ext\_lin\_b.png  
rotation.png  
stride.png

Návod k použití a popis experimentů:

README

Tento text bakalářské práce:

xcerno19.pdf

xcerno19.odt