

NÁVRH A OPTIMALIZACE OBRAZOVÝCH KLASIFIKÁTORŮ

Filip Kadlček

VTI-DR-4, 2. ročník, denní studium

Vedoucí: Otto Fučík

FIT VUT v Brně

Božetěchova 2, Brno, Česká Republika

ikadlcek@fit.vutbr.cz

Abstrakt. Práce se zabývá problematikou obrazových klasifikátorů a jejich efektivní syntézou do FPGA technologie. V práci je uveden postup pro automatickou syntézu klasifikátoru na základě trénovacích a testovacích dat. Automatická syntéza klasifikátoru je vykonávána pomocí iteračního procesu. Řada dílčích výsledků iteračního procesu je uvedena v práci.

Klíčová slova. AdaBoost, Local Binary Pattern, FPGA, Iterační návrh, Tvar příznaků.

1 Úvod

Obrazové klasifikátory se používají ve velkém množství aplikací jako jednotky pro předzpracování obrazu. Slouží k označení významných míst v obraze, na kterých se provede případně další zpracování. Příkladem takových aplikací jsou například systémy pro detekci objektů v obraze a různé dohledové a bezpečnostní systémy. Obrazové klasifikátory se tak stávají aplikací, která stojí v přední linii při zpracovávání obrazových dat. Výsledek klasifikace tak má velký vliv na aplikace, které zpracovávají pouze data, jež jsou označena klasifikátorem jako oblasti zájmu. Klasifikátory významně ovlivňují celou aplikaci a její výslednou výkonnost. Zpravidla se výkonnost klasifikátorů posuzuje jen dle klasifikační úspěšnosti, tedy jak je klasifikátor úspěšný v detekci rozpoznávaných předmětů.

Existuje však ještě další metrika, která je v mnoha aplikacích zanedbávána – rychlost klasifikátoru. Při zpracování obrazových dat velmi často potřebujeme, aby aplikace pracovala v reálném čase na daném toku dat. Klasifikace je zpravidla velmi náročná aplikace, která zkonsumuje velmi velký výpočetní výkon. Na další zpracování obrazu tak již nezůstane mnoho prostoru. Z tohoto důvodu je vhodné konstruovat klasifikátory tak, aby spotřebovali co nejméně výpočetního výkonu a pracovali velmi efektivně. Běžný procesor neposkytuje většinou dostatečný výkon ke zpracování obrazu ve vysokém rozlišení (1920 x 1200 pixelů). Poté je nutné přistoupit k akcelerování výpočtu pomocí speciálních prostředků, jako je například FPGA technologie (Field-Programmable Gate Array).

Dosavadní práce v oblasti obrazových klasifikátorů se zabývaly vždy buď jen trénováním vybraného klasifikátoru, anebo jen implementací natrénovaného klasifikátoru do FPGA. Obě oblasti tak byly do jisté míry rozděleny. Tato práce se zabývá možností propojení těchto dvou zdánlivě odlišných oblastí. Doposud se optimalizoval proces návrhu klasifikátoru do FPGA vzhledem ke spotřebě zdrojů na FPGA až po získání natrénovaného klasifikátoru. Proces trénování klasifikátoru však nebral v úvahu, že výsledný klasifikátor bude implementován na FPGA. Hlavním cílem práce je tak vytvoření metodiky, která pomocí iteračního procesu, navrhne vysoce optimalizovaný klasifikátor pro libovolně zvolenou třídu problému a FPGA jednotku.

2 Návrh vysoce optimalizovaného klasifikátoru

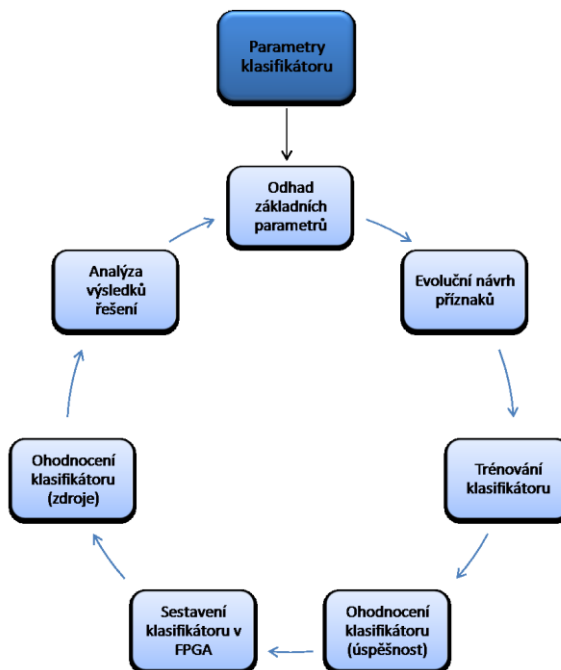
Při návrhu klasifikátoru se všechny dnešní techniky spoléhají na znalosti experta. Ten určí všechny důležité parametry klasifikátoru a následně se provede natrénování a sestavení požadovaného klasifikačního systému. Současný přístup při návrhu klasifikátoru do FPGA je tak prozatím rozdělen na dvě části - natrénování klasifikátoru a následné vytvoření klasifikátoru v FPGA z výsledků trénování. Trénovací algoritmus se vždy pokouší dosáhnout co nejlepších výsledků v klasifikaci a nebere ohledy na možnou syntézu v FPGA. Při sestavování klasifikačního systému v FPGA zpravidla již není možné pozměnit klasifikátor a tak se pouze vytvářejí konstrukce, jak co nejeefektivněji implementovat daný klasifikátor v FPGA.

Každá z uvedených částí návrhu klasifikátoru v FPGA řeší zcela odlišnou disciplínu. Cílem práce je odstranit bariéru mezi uvedenými disciplínami. Výsledkem bude algoritmus, který provádí iterační návrh klasifikátoru pro FPGA a automatizovaně nalezne optimalizované řešení dle zadaných vstupních podmínek. Architektura pro návrh vysoce optimalizovaného klasifikátoru v FPGA již byla představena v [3], [4]. V uvedených pracích jsou popsány možnosti optimalizace klasifikátoru a základní možnosti propojení algoritmu pro syntézu klasifikátoru do FPGA s trénovacím algoritmem klasifikátoru.

2.1 Iterační schéma syntézy klasifikátoru

Klasifikační systém má mnoho parametrů, podle kterých se optimalizuje. Tyto parametry jsou popsány v následující sekci 3. Každý parametr má pro klasifikační systém jiný význam a pro každý parametr je třeba sestavit cenovou funkci, která nám bude popisovat, jak dobře plníme požadavky daného parametru. Obrázek 1 zobrazuje základní schéma iteračního procesu návrhu klasifikátoru. Jednotlivé položky ze schématu jsou postupně vysvětleny:

- *Parametry klasifikátoru:* Při návrhu klasifikátoru máme několik požadavků, které stojí mimo iterační proces návrhu klasifikátoru. Takovým parametrem je například předem stanovená úspěšnost klasifikátoru, počet zdrojů v FPGA, které nesmí výsledný klasifikátor přesáhnout, a také vstupní trénovací a testovací data, na kterých bude probíhat trénování a testování.
- *Odhad základních parametrů:* Tím se rozumí například velikost detekčního okna v FPGA.



Obrázek 1: Iterační proces návrhu klasifikátoru

- *Evoluční návrh příznaků:* Problematika je popsána v kapitole 4.
- *Trénování klasifikátoru:* Úloha představuje natrénování klasifikátoru dle stanovených vstupních parametrů.
- *Ohodnocení klasifikátoru:* Získání klasifikační úspěšnosti daného řešení.
- *Sestavení klasifikátoru v FPGA:* Natrénovaný klasifikátor se syntetizuje pro FPGA technologii. V [3] jsou představeny základní možnosti, jak je možné z procesu sestavení klasifikátoru získat parametry pro trénovací proces. Výstupem tohoto procesu je především seznam příznaků, které mohou být použity, avšak jejich použitím nenastane příliš velký nárůst zdrojů.
- *Ohodnocení klasifikátoru:* Po sestavení klasifikátoru pro FPGA se provede jeho ohodnocení vzhledem k požadavkům na spotřebovávané zdroje.
- *Analýza výsledků:* V této fázi se provede analýza nad výsledky získanými trénováním a syntézou klasifikátoru. Výsledkem této fáze je úprava parametrů trénovacího procesu klasifikátoru.

Proces syntézy výsledného klasifikátoru probíhá v iteracích a s každou iterací se přibližuje k požadovanému výsledku. V reálné implementaci se předpokládá, že jednotlivé části iteračního procesu mohou mezi sebou komunikovat přímo a je tak redukován celkový počet iterací. Proces trénování klasifikátoru se může například pozastavit a požádat o předběžnou analýzu, zda aktuální klasifikátor stále splňuje požadované meze nebo ne. Tato analýza však není zcela přesná, ale poslouží jako vhodná nápověda, zda pokračovat v trénovacím procesu, nebo zda jej ukončit.

Iterační proces je možné optimalizovat, například některou z technik evolučních algoritmů (horolezecká metoda). Optimalizační metoda pomůže ke zrychlení návrhu a nalezení vhodných parametrů klasifikátoru, dle stanovených požadavků, ale i k nalezení výsledného klasifikátoru. V procesu optimalizace klasifikátoru se použije více-kriteriální fitness funkce. Metodika pro návrh klasifikátorů v FPGA je navržena dostatečně obecně tak, aby pracovala pro širokou množinu klasifikačních algoritmů. Pro ověření funkčnosti je zvolen algoritmus AdaBoost [2], [8], [9], jelikož má velmi dobré vlastnosti vzhledem k implementaci výsledných klasifikátorů v FPGA.

3 Nalezení vhodných parametrů klasifikátoru

AdaBoost má několik základních parametrů, které ovlivňují jeho výkonnost – velikost detekčního okna a typ použitých slabých klasifikátorů. Každý z těchto parametrů se musí určit zvlášť pro každou klasifikační úlohu. V této sekci jsou představeny parametry klasifikátoru, které jsou zkoumány v rámci optimalizačního procesu. Doposud se všechny práce zabývaly jen parametry, které ovlivňují účinnost klasifikátoru. Tato práce jde ve zkoumání parametrů dále a zabývá se, mimo jiné, i vlivem parametrů na výsledný návrh klasifikátoru v FPGA. Kapitola předpokládá znalost problematiky AdaBoost [8], [9] a LBP [7]. Následně jsou představeny doposud neobvyklé parametry pro klasifikátor jako je účinnost klasifikátoru a velikost výsledného klasifikátoru v FPGA. Tyto parametry se mohou zdát nesmyslné, jejich význam je však opodstatněn. Zkoumané parametry jsou:

- **Velikost detekčního okna:**

Parametr je velmi důležitý. Detekční okno je oblast obrazu, která se zkoumá v jednom běhu klasifikátoru, určuje tak i velikost hledaného předmětu v obraze. Pro prohledání celého obrazu se provádí několika násobný průchod obrazem daným oknem. Velikost detekčního okna nám zpravidla určuje, kolik detailů v obraze se rozpoznává klasifikátorem. Velké detekční okno vede na příliš specifický klasifikátor, který však ztratí na své obecnosti klasifikovat předměty obecného typu (například automobil). Velikost detekčního okna je také velmi úzce spjata s počtem zdrojů potřebných pro implementaci daného klasifikátoru v FPGA (větší okno – více zdrojů).

- **Tvary příznaků slabých klasifikátorů:**

Obecně navržené příznaky nemusejí být vždy tím nejlepším řešením pro všechny typy úloh.

V práci je navržen optimalizační postup, který navrhne příznaky pro každou aplikaci zvlášť. Při návrhu nových příznaků je současně kladen i velký důraz na implementaci příznaků v FPGA a již při procesu návrhu příznaků jsou dodržována potřebná kritéria.

- **Účinnost klasifikátoru:**

Zpravidla se nepovažuje za jeden ze vstupních parametrů pro klasifikátor, jelikož se zatím všechny práce pokoušely dosáhnout co nejlepší účinnosti klasifikátoru a dle účinnosti pak posuzovaly kvalitu svého řešení. Může však nastat situace, kdy účinnost považujeme za jeden ze vstupních parametrů klasifikace. Tato situace nastává, pokud potřebujeme natrénovat co nejmenší klasifikátor, který však bude splňovat zadanou účinnost.

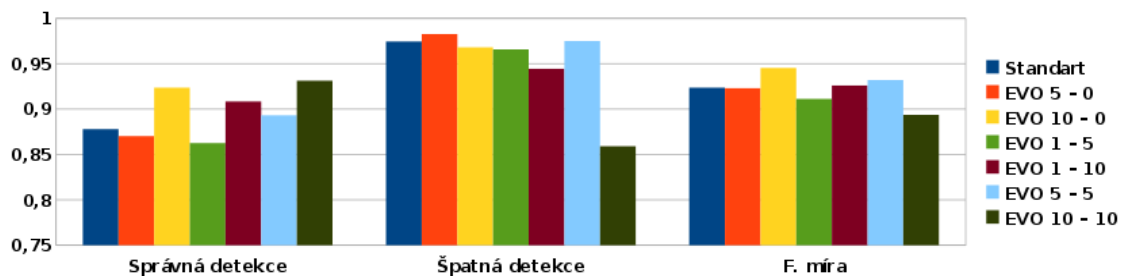
- **Velikost klasifikátoru v FPGA:**

Tento parametr se v dosavadních trénovacích algoritmech nevyskytuje. Současný přístup v konstrukci klasifikátorů pro FPGA je takový, že se vezme již natrénovaný a ověřený klasifikátor, který je však sestaven pro použití v běžném PC, a poté je vytvořena jeho implementace v cílové architektuře, kterou může být například FPGA. Výsledná velikost návrhu tak není žádným způsobem přivedena na vstup trénovacího algoritmu klasifikátoru. Tato situace nastává především proto, že se proces trénování a následného sestavení klasifikátoru provádí pouze jedenkrát a to zpravidla pouze manuální formou sestavení. Vytvoříme-li funkci, která provádí odhad velikosti klasifikátoru v procesu trénování, můžeme jako jeden ze vstupních parametrů klasifikace používat velikost výsledného návrhu pro FPGA technologii.

4 Návrh optimalizovaných tvarů příznaků LBP funkcí

V kapitole je popsán návrh nových tvarů příznaků pro klasifikátory založené na Local Binary Pattern – LBP funkcích (případně MB-LBP [6]). V současných implementacích klasifikátorů, které využívají LBP funkce, se používají jen základní tvary masek (matic) pro výpočet příznaků. Tyto základní tvary byly navrženy ručně a nevyužívají žádný potenciál možné optimalizace vzhledem k vybrané třídě problémů. Původní příznaky sice pracují v širokém spektru úloh velmi dobře, ale ani pro jednu úlohu nejsou optimální. Zde je představena možnost, která provádí návrh tvarů příznaků pro řešení jednoho konkrétního problému. V takovém přístupu je velká šance, že se nám podaří navrhnout nový a zcela revoluční tvar příznaku. Ten bude ve vybrané úloze pracovat velmi dobře, ale pro jiné úlohy může být zcela nepoužitelný.

Se získáním nového tvaru příznaku, který bude vykazovat dobré výsledky při klasifikaci, se nám podaří zvýšit účinnost celého silného klasifikátoru. Další výhodou tohoto přístupu je, že pro vybudování silného klasifikátoru s požadovanou účinností budeme potřebovat nižší počet slabých klasifikátorů. To je velmi žádoucí, pokud budeme výsledný silný klasifikátor implementovat v FPGA. Sníží se tak ještě více požadavky klasifikátoru na zdroje v FPGA. Detailní popis implementace je uveden v článku [5]. Pro návrh nových tvarů příznaků byla vybrána evoluční metoda. V následujícím grafu jsou znázorněny výsledky klasifikátorů, které pracují právě s využitím nově navržených příznaků.



Obrázek 2: Porovnání výsledků AdaBoost klasifikátorů založených na standardních a evolučních příznacích

Výsledky na obrázku 2 byly získány na testovacím datasetu UIUC [1]. Obrázek 2 zobrazuje výsledky silných klasifikátorů založených pouze na standardních tvarech příznaků a výsledky silných klasifikátorů založených na evolučně navržených tvarech ve spojení se standardními tvary. Označení klasifikátorů *EVO 10 – 5* značí, že silný klasifikátor je sestaven ze tří částí. Celý klasifikátor má celkem 100 slabých klasifikátorů. Číslice 10 v označení klasifikátoru značí, že úvodní část klasifikátoru je složena z 10 slabých klasifikátorů, jež využívají evolučně navržené tvary příznaků. Číslice 5 značí, že závěrečná část klasifikátoru je složena z 5 slabých klasifikátorů, jež využívají evolučně navržené tvary příznaků. Zbylé slabé klasifikátory (85) jsou založeny na standardních tvarech příznaků.

Obrázek 2 má tři hlavní části – správná detekce, špatná detekce a F míra. Všechny hodnoty grafu jsou uvedeny v procentech. F míra je kombinace správné detekce a špatné detekce. Z obrázku je možné vidět, že klasifikátor *EVO 10 – 0* dosahuje nejlepších výsledků. Tento klasifikátor vykazuje zlepšení v přesnosti klasifikace oproti standardnímu přístupu o cca 4 %. Procento špatných klasifikací je téměř stejné v porovnání se standardním přístupem. Nesprávné použití evolučně navržených příznaků může vést ale i ke zhoršení klasifikace, odůvodnění je možné naleznout v [5].

5 Závěr

Práce je zaměřena na automatizovanou syntézu vysoce optimalizovaných klasifikátorů pro FPGA technologii a pokouší se vytvořit obecný postup pro sestavení optimalizovaného klasifikátoru. Demonstrace řešení daného problému je provedena na klasifikátoru AdaBoost. Dílčí výsledky práce jsou prezentovány v pracích [3], [4] a [5]. Pro syntézu vysoce optimalizovaného klasifikátoru je v práci použita iterační metoda, jež sestává z několika dílčích částí. Vybrané dílčí části jsou v práci krátce popsány. Dále jsou představeny nové parametry pro trénování klasifikátorů, jako je například parametr udávající velikost zabraného místa v FPGA. Ač jsou uvedené parametry nevýznamné při implementaci běžným způsobem, při implementaci navrhovanou metodou mají své opodstatnění a je třeba s nimi pracovat při návrhu klasifikátoru.

Další část iteračního procesu, jež je krátce uvedena, popisuje nalezení optimálních tvarů příznaků pro LBP klasifikátory. Z výsledků je patrné, že budeme-li provádět návrh nových příznaků vždy jen pro zadaný problém, tak nově navržené příznaky budou vykazovat lepší vlastnosti a jejich pozitivní vliv se promítne na celý silný klasifikátor. Díky tomuto řešení dosáhneme při použití stejného počtu slabých klasifikátorů vyšší účinnosti silného klasifikátoru (zlepšení o cca 3 – 4 %), případně pro dosažení specifické účinnosti silného klasifikátoru budeme potřebovat menší počet slabých klasifikátorů. Nově navrhované příznaky je také možné optimalizovat pro použití v FPGA a snížit tak požadavky na implementaci výsledného klasifikátoru v FPGA. Vybrané části iteračního algoritmu již byly implementovány. Jedná se především o části: evoluční návrh nových příznaků, trénovací algoritmus AdaBoost a automatická syntéza klasifikátoru v FPGA.

5.1 Stanovení hypotéz

V práci byly stanoveny následující hypotézy pro disertační práci:

1. *Je možné automatizovaně navrhnout vysoce optimalizovaný silný klasifikátor pro FPGA technologii, který bude vykazovat srovnatelné výsledky klasifikace, jako klasifikátory navržené konvenčními metodami.*
2. *Aplikačně specificky navržené tvary příznaků pro slabé klasifikátory dosahují vyšší účinnosti při klasifikaci než klasifikátory založené na konvenčních tvarech příznaků a současně mohou být navrženy vhodně pro FPGA technologie.*

5.2 Postup prací a cíle disertační práce

Následující práce budou zaměřeny především na sestavení dílčích částí iteračního procesu syntézy klasifikátoru. To je integrace již existujících částí a vytvoření rozhraní mezi všemi částmi. Po dokončení iteračního trénovacího algoritmu se předpokládá ověření implementace v FPGA. Z tohoto důvodu bude sestavena testovací platforma, na které bude rovněž probíhat ověřování na reálných datech.

Pro úspěšné dokončení disertační práce byla stanovena řada cílů, které jsou postupně plněny:

- Vymezení hlavních parametrů pro AdaBoost klasifikátor.
- Vytvoření algoritmu pro odhad základních parametrů AdaBoost klasifikátoru.
- Návrh nových aplikačně specifických tvarů příznaků pro slabé klasifikátory.
- Vytvoření trénovacího algoritmu AdaBoost, který bude mít vstupy pro ovlivnění výsledného klasifikátoru vzhledem k FPGA implementaci.
- Vytvoření nebo využití existujícího algoritmu pro hodnocení úspěšnosti silných klasifikátorů a jeho integrace do celého procesu.
- Vytvoření algoritmu pro automatickou syntézu silného klasifikátoru do FPGA s výstupem pro trénovací algoritmus AdaBoost klasifikátoru.
- Vytvoření funkce pro ohodnocení silného klasifikátoru implementovaného v FPGA dle hledisek úspěšnosti a spotřeby zdrojů.
- Vytvoření platformy pro běh klasifikátorů a otestování klasifikátorů na reálných datech.

Hlavním cílem práce je nalezení algoritmu, který bude syntetizovat silně aplikačně specifické klasifikátory, které však budou dosahovat lepších výsledků než doposud publikované klasifikátory.

Poděkování

Tato práce byla vypracována v rámci projektu Centrum excelence IT4Innovations, reg. č. CZ.1.05/1.1.00/02.0070 podporovaného Operačním programem Výzkum a vývoj pro inovace, financovaného ze strukturálních fondů EU a ze státního rozpočtu ČR.

Literatura

- [1] Agarwal, S., Awan, A., Roth, D.: *UIUC Image Database for Car Detection*.
- [2] Freund, Y., Schapire, R.: *A short introduction to boosting*. *Soc. for Artif.*, no. 5, 1999: p. 771-780.
- [3] Kadlček, F.: *Implementace obrazových klasifikátorů v FPGA*. FIT VUT v Brně, diplomová práce, Brno, 2010.
- [4] Kadlček, F., Zemčík, P., Juránek, R.: *Automatic synthesis of classifiers in FPGA*. In *International Bata conference for Ph.D. Students and Young Researchers*, Tomas Bata University in Zlin, 2011, ISBN 978-80-7454-013-4.
- [5] Kadlček, F., Fučík, O.: *Evolutionary design of Local Binary Pattern feature shapes for object detection*. In *NASA/ESA Conference on Adaptive Hardware Systems (AHS-2012)*, Nuremberg, Germany, 2012, s. 8.
- [6] Liao, S., Zhu, X., Lei, Z., et al.: *Learning Multi-scale Block Local Binary Patterns for Face Recognition*. *Chinese Academy of Sciences*, China, 2007, p. 828-837.
- [7] Mäenpää, T.: *The Local Binary Pattern approach to texture analysis – extensions and applications*. PHD thesis, Faculty of Technology, University of Oulu, 2003.
- [8] Viola, P., Jones, M.: *Real-time object detection*. *Int. J. Comput. Vision*, vol. 57, no. 2, p. 137–154, May 2004.
- [9] Viola, P. and Jones, M.: *Rapid object detection using a boosted cascade of simple features*. In *CVPR*, 2001.