



**FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI**

**KATEDRA
GEOMATIKY**

Zpracování digitalizovaného obrazu (KKY/ZDO)

Závěrečná zpráva

Incision quality evaluation

Tomáš Andrš & Jan Hromada

V. ročník

Datum zpracování: 15. 05. 2023

Zadání

“Cílem této práce je zhodnotit kvalitu chirurgického stehu na základě obrazu řezu a stehu. Studentům bude poskytnuta sada obrázků zobrazujících řezy a stehy provedené během chirurgického zákroku. Úkolem studentů bude pomocí metod počítačového vidění extrahovat a segmentovat obrazová data a následně pomocí metod strojového učení provést kvalitativní analýzu stehu. Studenti budou hodnotit kvalitu stehu na základě parametrů jako je rovnost a kolmost. Výsledky analýzy budou prezentovány pomocí grafických vizualizací a metrik kvality šití.”

Vysegmentovaná obrazová data je třeba následně převést do formátu JSON a výstup celé práce nahrát na úložišti GitHub obsahující zejména závěrečnou zprávu a funkčně navržený program.

Vstup

Vstupními daty je pro nás sada celkem 220 RGB snímků o různém rozlišení zachycující (většinou) jednu incizi a rozdílný počet stehů na kůži prasečí nohy. Tyto snímky byly následně anotovány pomocí Computer Vision Annotation Tool (CVAT) studenty předmětu KKY/ZDO v letošním roce 2022/2023.

Spuštění programu

V rámci studia jsme se ještě s tvorbou takového programu ani s odevzdáním prostřednictvím úložiště GitHub nesetkali, a tak doufáme, že jsme postupovali správně. Na [GitHub](#) jsme nahráli následující soubory:

- `run.py`
- `output.json`
- `image001.jpg ... image220.jpg`
- `Zaverecna_zprava.pdf`
- `processed.json` - (JSON obsahující výsledky všech obrázků)

Dle zadání umožňuje program dvě možnosti spuštění - s vizualizací a bez vizualizace. V obou případech je potřeba aby byly všechny soubory (tj. `run.py`, `output.json` a obrázky) v jedné složce. Před spuštěním je potřeba nainstalovat následující knihovny:

- `pip install scikit-image`
- `pip install matplotlib`

Pro spuštění programu bez vizualizace slouží následující příkaz:

```
> python run.py output.json image001.jpg image002.jpg
```

Po správném doběhnutí se do souboru `output.json` nahrají geometrie nalezených stehů a incize, dále také souřadnice průsečíků a jejich úhly pro dva zadané obrázky (obrázků může být více). Pro možnost spuštění s vizualizací slouží následující příkaz:

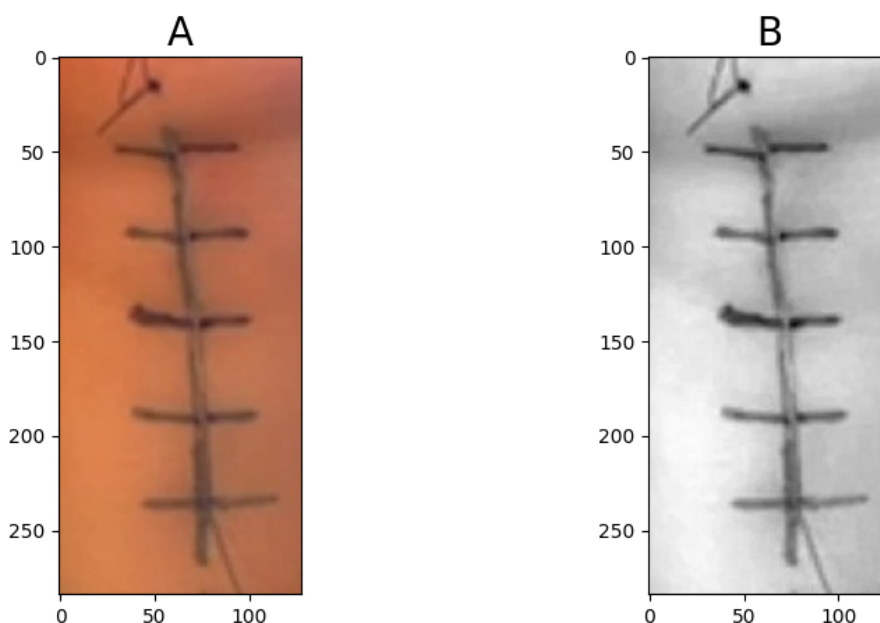
```
> python run.py output.json -v image001.jpg image002.jpg
```

Po správném doběhnutí programu dojde k zobrazení výsledného obrázku i s geometriemi stehů a incize ve vyskakovacím okně. Do souboru `output.json` se opět nahrají výsledky a do složky se výsledné vizualizace uloží jako `.jpg` soubory s předponou `processed`.

Zpracování

Převedení vstupních dat do šedotónu

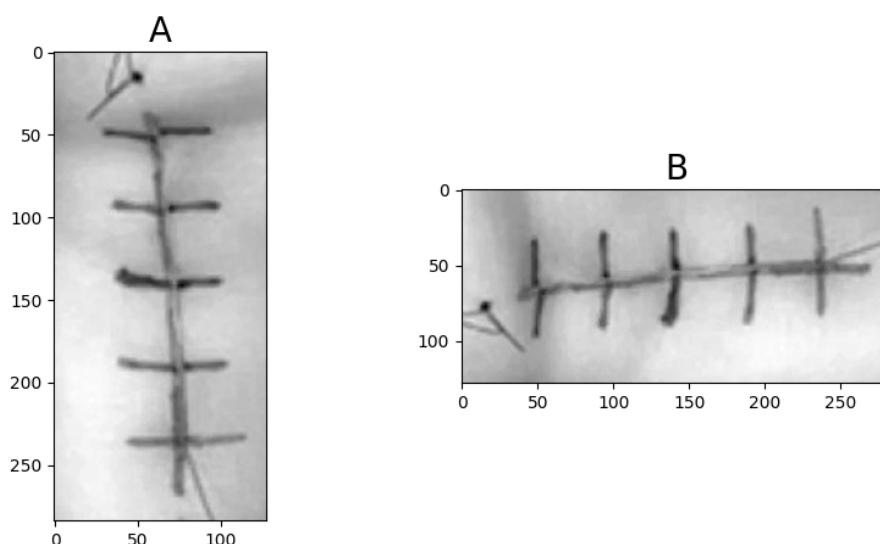
V rámci následujících kroků, zejména kvůli segmentaci, je třeba pracovat s šedotónovým obrázkem. Při načtení obrázku jsme tedy převedli RGB obrázek do šedotónu. Výsledek po této operaci je zobrazen na následujícím obrázku 1.



Obrázek 1: Převedení vstupního snímku do šedotónu. A - před převodem, B - po převodu.

Kontrola natočení

Na počátku zpracování jsme v jedné z přidělených sad snímků narazili na snímek, který byl orientovaný na výšku, a proto jsme se rozhodli provést kontrolu natočení a v případě, že je obrázek orientován na výšku, dojde k jeho otočení. (Ve finálním datasetu 220 snímků jsme doposud objevili pouze jeden obrázek, který je takto natočený.) Výsledek této operace je zobrazen na následujícím obrázku 2.



Obrázek 2: Natočení snímku.

Segmentace

Už v návrhu na zpracování semestrální práce jsme zmiňovali, že největším problémem v navrženém algoritmu je právě segmentace. Již od počátku zpracování této práce jsme zkoušeli různé druhy segmentace, jako například vyhlazování histogramu pro hledání optimálního prahu podle minima, nebo vyhledání prahu podle Otsuovi metody. Žádná z těchto metod se však pro zkušební sadu obrázků nejevila jako správná a s jejich využitím nebyl následný proces vyhledávání geometrie funkční. Alternativně jsme zkoušeli obraz nejprve převést pomocí Fourierovi transformace, kde jsme na signál využili Low pass metodu. Výsledný převedený obraz, který jsme chtěli dále segmentovat, však nejevil dobré výsledky.

Z toho důvodu jsme se rozhodli využít metodu Watershed z knihovny skimage. Metoda bere jako vstup výsledek hranového detektoru aplikovaného na vstupní obrázek a dále tzv. Markers, které vzniknou vyprahováním původního obrázku pomocí dvou prahů. Nejprve jsme hodnotu těchto prahů nastavili ručně a na obrázku zkoušeli další postup. Následně jsme se i zde snažili využít automatické hledání prahu pomocí Otsuovi metody. Po zkoušce na více obrázcích byla však i tato metoda nedostačující.

V jednom z mnoha pokusů jsme vyzkoušeli vylepšit segmentaci definováním pozadí snímku pomocí rohových pixelů. V těchto rozích jsme použili okno o velikosti závislé na rozměrech obrázku, ve kterých jsme předpokládali, že se nebudou vyskytovat žádné řezy ani stehy. Cílem bylo nalézt ideální práh, který by reprezentoval pozadí. Tento nalezený práh jsme poté využili k definování pozadí v Markers (viz výše zmíněno). Bohužel, segmentace pomocí tohoto postupu nedosáhla očekávaného výsledku, protože některé obrázky obsahují i jiné objekty než jen prasečí nohu (viz obrázek 3). Kromě rohových pixelů jsme zkoušeli i celé pásy ve vrchní a spodní části obrázku. Žádná z těchto metod však nevedla ke zdárné segmentaci. Předpokládáme, že pokud by byly světelné podmínky při pořizování fotografií konstantní, mohla by tato metoda vykazovat lepší výsledky.

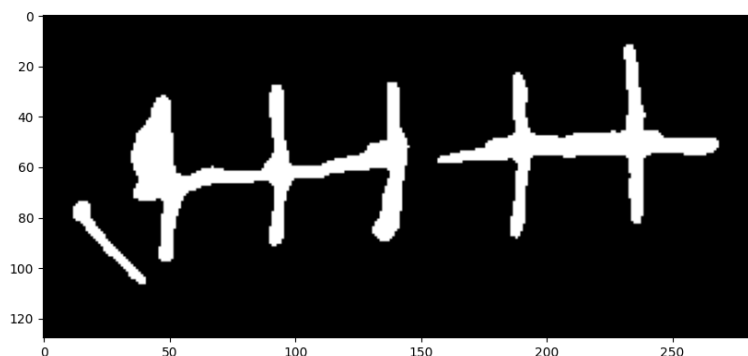


Obrázek 3: Příklad rozdílných hodnot v rozích.

Výsledné metody, které jsme pro segmentaci obrazu využili je procentní prahování. Zůstali jsme však u metody Watershed, kde hodnotu prahů určujeme v závislosti na procentuálním rozložení pozadí a objektu (stehu s incizí) v obrázku. Hodnota těchto prahů se hledá iterativně a kritériem pro ukončení je procentuální rozložení objektu a pozadí. Tyto procenta jsme odvodili na základě průměrných hodnot z obrázků, které jsme měli k dispozici (jedná se o stejné obrázky, které jsme měli jako skupina za úkol anotovat). Tímto postupem jsme se zbavili ručního nastavení prahu. Jako počáteční hodnota se využívá již zmiňovaný práh nalezený pomocí Otsuovi metody. Tato metoda jeví dobré výsledky pro obrázky s relativně konstantním pozadím. Pokud jsou však v obrázku rozdílné jasové podmínky (tmavé, nebo naopak výrazně světlé části) obrázek se vysegmentuje špatně, což ovlivní i celkové odhalení stehů a incize v obrázku. Metoda jeví ve většině případů lepší podmínky, pokud je před segmentací využita filtrace pomocí Gaussovského filtru, a proto jsme se ji rozhodli před proces segmentace zakomponovat.

Jak bylo zmíněno výše, takto navržený postup jeví poměrně dobré výsledky pro obrázky s konstantními jasovými podmínkami. Ty ale nejsou při pořizování fotografií často zajištěny. Další nežádoucí vlastností je značně rozdílné rozlišení vstupních obrázků. V obrázcích s malým rozlišením zabírají stehy a incize většinou větší část obrazu, čímž nedojde ke správnému výběru Markers a tím tedy i správné segmentaci. Jsme si tedy vědomi toho, že se nejedná o ideální postup, nedokázali jsme však nalézt jinou metodu, která by poskytovala lepší výsledky, přestože

jsme nad procesem segmentace strávili téměř polovinu času zpracování semestrální práce. Výsledek segmentace je zobrazen na následujícím obrázku 4.

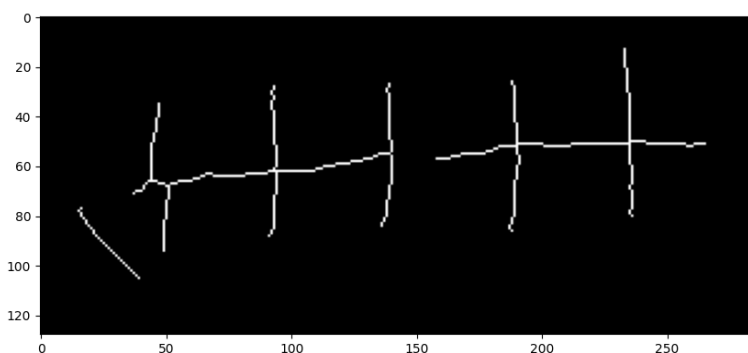


Obrázek 4: Výsledek segmentace.

Vytvoření skeletu

Jak bylo popsáno v návrhu, dalším krokem byla aplikace morfologických operací na vysegmentovaný obrázek z předchozího kroku. Na úvod jsme zkoušeli využít erozi, pomocí které jsme se snažili odstranit nežádoucí objekty ve vysegmentovaném obrázku. Mezi takovéto objekty patří převážně tečky, kterými si zřejmě studenti medicíny, kteří provádí zašití, naznačují, kudy má steh procházet. Tento postup však v některých případech odstraňoval i objekty, které jsou naopak důležité (stehy/incize). Kromě eroze jsme zkoušeli aplikovat i uzavření a otevření - žádná z těchto operací však nevykazovala dobré výsledky.

Morfologická operace, kterou jsme nakonec využili, je vytvoření skeletu, pomocí kterého dostaneme (pro dobře vysegmentovaný obrázek) přibližný tvar výsledných stehů a incize. Skelet je zároveň vhodný pro další postup. Výsledek vytvoření skeletu je zobrazen na následujícím obrázku 5.

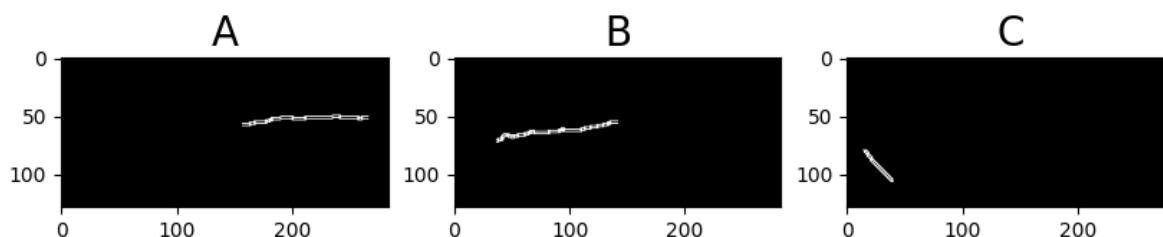


Obrázek 5: Vytvoření skeletu.

Labelling

K jednoznačnému rozdělení jednotlivých částí skeletu obrázku jsme využili metodu labeling. Pomocí této metody je možné přistupovat ke každému souvislému úseku skeletu zvlášť, spočítat počet objektů v obraze nebo identifikovat nežádoucí úseky, které můžeme následně odstranit. Jelikož při odhalování geometrie (viz *Detekce objektů pomocí Houghovi transformace*) přistupujeme ke každému podezřelému objektu zvlášť, je nejprve potřeba jednotlivé objekty na obraze detekovat a následně očíslovat. Jednotlivé objekty jsme ve skeletu odhalili aplikací hranového detektoru v obou směrech. Ve směru vodorovném je odhalena incize (ta by měla být pouze jedna, při zpracování jsme však narazili i na obrázek obsahující dvě incize) a ve směru kolmém jsou odhaleny jednotlivé stehy. Následně jsme využili metodu Label z knihovny skimage, která očíslovuje všechny objekty v obraze.

Při aplikaci hranového detektoru jsou často odhaleny i nežádoucí šumy, které za stehy či incizi nelze považovat a je potřeba vyřešit jejich odstranění. Šumy jsme vyfiltrovali počítáním velikosti jednotlivých objektů v obou směrech. Pokud je objekt menší než definovaná hodnota, je vyfiltrován (odstraněn). Oproti návrhu jsme zde přidali výběr hodnoty podle velikosti právě zkoumaného obrázku. Výsledek nalezených objektů pomocí labellingu a kontroly velikosti objektu je zobrazen na obrázku 6.



Obrázek 6: Výsledek po detekci objektů v jednom směru (hledání incize). Do objektů je zařazen i nežádoucí objekt (C), který vznikl detekcí části nitě, kterou student na obrázku zanechal.

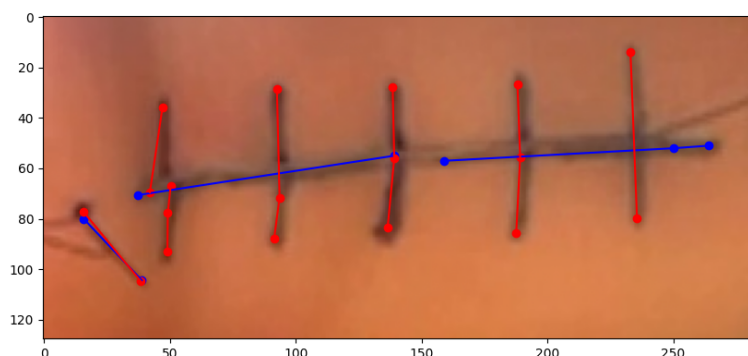
Detekce objektů pomocí Houghovi transformace

Na všechny objekty nalezené v předchozím kroku byla následně aplikována Houghova transformace, jejímž úkolem je nalezení bodů, které dobře aproximují stehy a incize. Tyto body vzniknou jako průsečíky přímk nalezených Houghovou transformací.

Pro každý objekt je vytvořena tabulka odhalených přímek. Tato tabulka se pak prochází a hledají se průsečíky mezi jednotlivými přímkami. Pokud je průsečík nalezen uloží se do seznamu podezřelých průsečíků. Dalším postupem je kontrola podezřelých průsečíků. Pokud se průsečík nachází na daném objektu, je uložen do výsledného seznamu průsečíků pro daný objekt. Nalezené průsečíky by měly aproximovat tvar nalezeného objektu.

Pomocí předchozího postupu jsou nalezeny vnitřní body objektů, ještě však zbývá určit počáteční a koncový bod daného objektu. Ten je určen podle souřadnic snímku, tj. v případě incize se seřadí pixely objektu podle souřadnice y (vodorovná) a vybere se první a poslední souřadnice. Oproti návrhu jsme zde přidali průměrování na prvních a posledních 10 pixelů objektu. Nalezením počátečního a koncového bodu jsme dostali souřadnice bodů výsledného objektu.

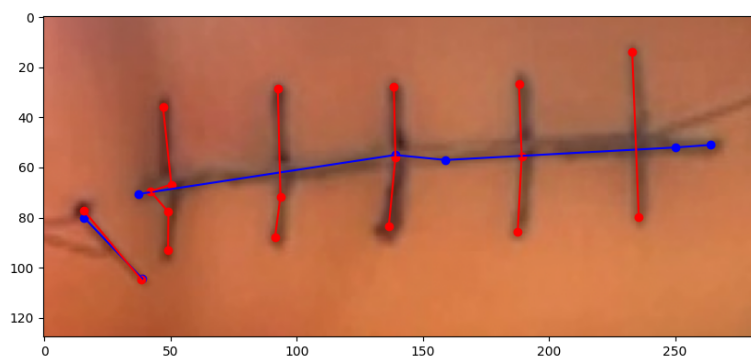
Celý postup je opakován pro všechny objekty v obou směrech. Výsledek po nalezení všech objektů je zobrazen na následujícím obrázku 7.



Obrázek 7: Výsledek po nalezení objektů pomocí Houghovy transformace. Červenou barvou jsou znázorněny nalezené stehy, modrou barvou incize.

Spojení detekovaných objektů

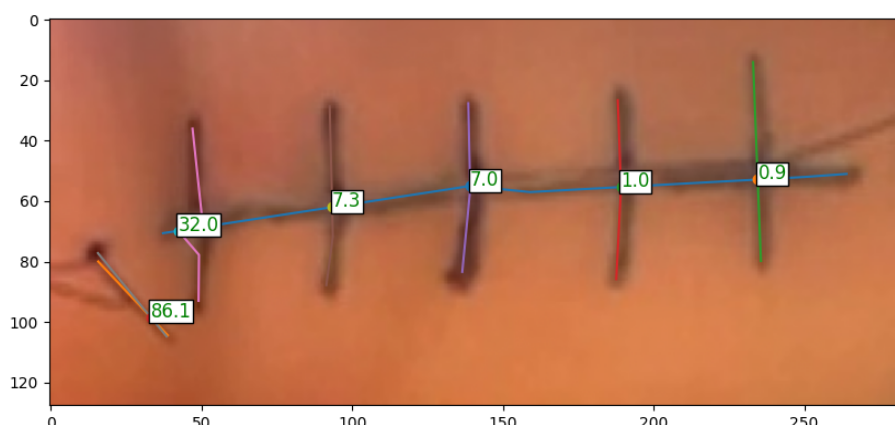
Výsledkem předchozího kroku jsou liniové úseky skládající se z několika segmentů. Tyto liniové úseky reprezentují (v ideálním případě) incizi a stehy. Při zkoušení algoritmu na více obrázcích, jsme však objevili problém. Některé objekty (převážně stehy) byly detekovány jako dva objekty, přestože se jednalo pouze o jeden, tj. steh nad incizí a pod incizí byl detekován jako dva objekty, přestože se ve skutečnosti jednalo o jeden steh. Problém jsme částečně vyřešili vytvořením funkce, která prozkoumává, jestli se v blízkosti bodů tvořících jeden objekt, nevyskytuje jiný objekt (musí se jednat o objekt stejného druhu, tj. ke spojení steh - incize nedojde). Pokud je nějaký objekt nalezen, dojde ke spojení těchto objektů. Výsledek po aplikaci spojení detekovaných objektů je znázorněn na následujícím obrázku 8.



Obrázek 8: Výsledek po provedení spojení blízkých objektů.

Detekce průsečíků stehů a incize, výpočet úhlu

Předposledním krokem bylo nalezení souřadnic průsečíků a úhlů mezi stehy a incizí. K tomu byl využit kód, který poskytl Ing. Zdeněk Krňoul, PhD. na úložišti Google Classroom předmětu ZDO. Kód byl mírně upraven tak, aby odpovídal našim výstupům. Výsledek po odhalení průsečíků a určení úhlů je znázorněn na následujícím obrázku 9.



Obrázek 9: Výsledek po určení průsečíků a úhlů mezi stehy a incizí.

Převod do JSON formátu

Poslední krok spočíval ve vytvoření výstupního formátu JSON podle šablony uvedené v zadání.

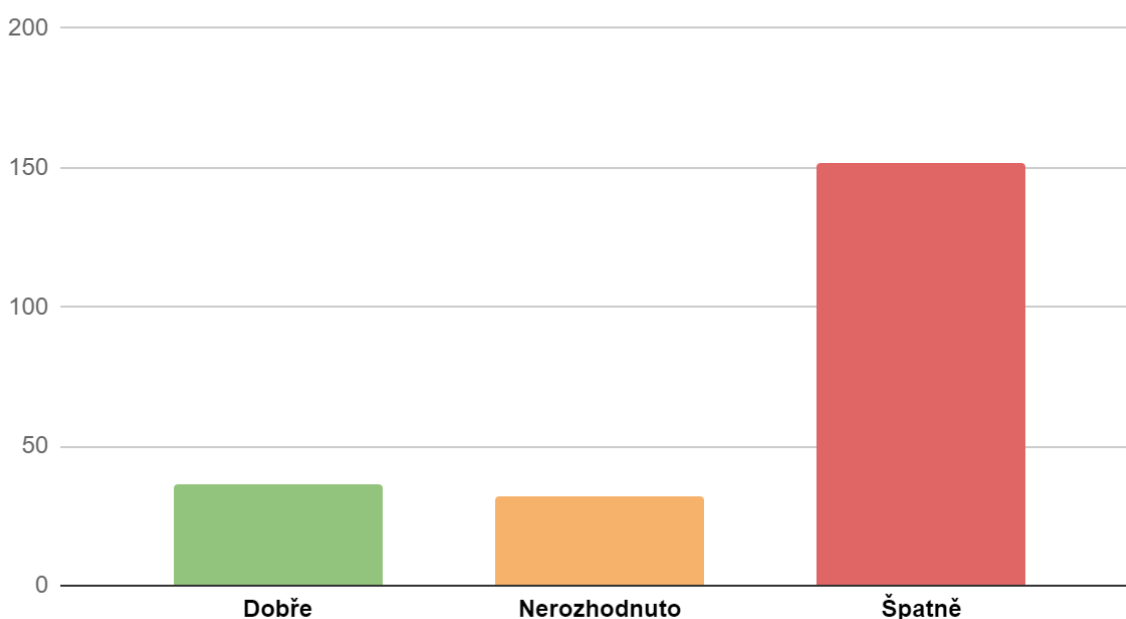
Analýza výsledků

Na závěr jsme provedli kontrolu funkčnosti navrženého algoritmu tak, že jsme jej aplikovali na celou sadu vstupních dat zahrnující 220 obrázků. Výsledky těchto obrázků jsou získány spuštěním přiloženého programu na úložišti GitHub, kdy se po jeho úspěšném doběhnutí uloží v podobě:

- processed_image001.jpg ... processed_image220.jpg

Nutno podotknout, že nejsme odborníky, a proto se jedná o neodbornou a čistě subjektivní analýzu, kterou je třeba brát s rezervou. Hodnocení jednotlivých snímků proběhlo na zvolené stupnici **Dobře** / **Nerozhodnuto** / **Špatně**. Stupni **Dobře** rozumíme obrázku, kde vyhledání incize a stehů proběhlo relativně úspěšně, avšak mnohdy nedokonale, jelikož se detekovala např. místa vpichu (černé tečky) nebo nitě vyčnívající z nich. Stupeň **Nerozhodnuto** představuje takové případy, kdy detekce incize proběhla úspěšně a detekce stehů neúspěšně a naopak. I zde se většinou setkáváme se šumem zmíněným ve stupni výše. Obrázky spadající do stupně **Špatně** jsou případy, kdy algoritmus selhal a detekce zájmových objektů mnohdy ani přibližně neodpovídala jejich skutečné poloze. Často se zde setkáváme se snímky o nízké kvalitě rozlišení a nerovnoměrném nasvícení. Výsledky jsme zaznamenali do tabulky a vytvořili graf viz obrázek 10, z něhož je patrné, že námi navržené řešení úlohy není ideální.

Úspěšnost algoritmu



Obrázek 10: Graf úspěšnosti navrženého algoritmu.

Závěr

Přestože jsme se snažili, algoritmus, který jsme navrhli, nenaplnuje požadavky této náročné úlohy. Hlavním důvodem selhání tohoto algoritmu je nedostatečná a také rozdílná kvalita vstupních dat, která často není ani lidským okem rozpoznatelná. Výsledky algoritmu

nasvědčují tomu, že z kvalitních vstupních snímků jsme schopni získat přijatelnou segmentaci v podobě incize a stehů. Bohužel, v opačných případech navržený algoritmus selhává.

Největší problém shledáváme v procesu segmentace. Jak jsme zjistili, automatizace oddělení pozadí od zájmových objektů nad více než jedním snímkem je obecně náročný úkol, zejména pak kvůli proměnlivosti vlastností každého vstupního snímku. V případě, že by snímky měly alespoň částečně podobné jasové vlastnosti, a byly v lepším rozlišení, bylo by možné proces segmentace zlepšit. Segmentaci jsme věnovali velkou část naší práce a prováděli nejvíce experimentů právě v tomto směru. I přesto jsme nedokázali přijít s lepším řešením.

Nevhodnost námi navržené segmentace pak ovlivňuje celkové výsledky hodnocení kvality stehů. Segmentace je jedním z prvních kroků, a pokud je snímek špatně nasegmentován, tak ani další kroky hodnocení výrazně nezlepší. Pokud je však segmentace aplikována na snímek podle ručně zadáných parametrů, jeví další kroky poměrně kvalitní výsledky.

Zpracování, pro které jsme se rozhodli, nám z hlediska znalostí o problematice zpracování digitálního obrazu dávalo zpočátku největší smysl. V průběhu studia této problematiky jsme se podrobněji dozvěděli o nejnovější a nejpoblárnější metodě v oblasti zpracování digitálního obrazu, kterou jsou neuronové sítě. Důvodem, proč jsme se touto cestou nevydali je fakt, že s metodou neuronových sítí nemáme kromě znalostí z tohoto předmětu, žádné zkušenosti. Pravdou také je, že jsme nechtěli, aby dosavadní postup, který jsme v tu chvíli již měli, přišel vniveč.