VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE A KLASIFIKACE VAD PANELŮ FOTOVOLTAICKÝCH ELEKTRÁREN Z TERMOKAMERY DRONU

DETECTION AND CLASSIFICATION OF PHOTOVOLTAIC POWER PLANT PANEL DEFECTS FROM A DRONE THERMAL IMAGING CAMERA

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — FILIP HAUŽVIC
AUTHOR

VEDOUCÍ PRÁCE — Ing. DANIEL BAMBUŠEK
SUPERVISOR

BRNO 2024

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Zadání bakalářské práce

154941

Ústav: Ústav počítačové grafiky a multimédií (UPGM)
Student: **Haužvic Filip**
Program: Informační technologie
Název: **Detekce a klasifikace vad panelů fotovoltaických elektráren z termokamery dronu**
Kategorie: Zpracování obrazu
Akademický rok: 2023/24

Zadání:

1. Prostudujte základy konvolučních neuronových sítí a jejich použití pro zpracování obrazu. Zaměřte se na práci s radiometrickými snímky.
2. Seznamte se stávajícími přístupy inspekcí fotovoltaických elektráren pomocí dronů a následné analýzy konkrétních vad na fotovoltaických panelech.
3. Vyberte vhodné metody a nástroje a navrhněte konvoluční neuronovou síť, která bude schopna detekovat a klasifikovat vady panelů FVE z radiometrických snímků.
4. Navrženou síť implementujte a natrénujte.
5. Proveďte experimenty a vyhodnoťte vlastnosti výsledného řešení.
6. Vytvořte video prezentující klíčové vlastnosti výsledného řešení.

Literatura:

- Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2nd, 2022
- Andriy Burkov: The hundred-page machine learning book, Large Print Book, 2019
- Goodfellow, Bengio, Courville: Deep Learning, MIT Press, 2016
- Gary Bradski, Adrian Kaehler: Learning OpenCV; Computer Vision with the OpenCV Library, O'Reilly Media, 2008
- Dále dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:
Body 1, 2, 3 a rozpracovaný bod 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Bambušek Daniel, Ing.**
Vedoucí ústavu: Černocký Jan, prof. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 9.11.2023

Abstrakt

Tato práce popisuje proces zpracování termografických snímků fotovoltaické elektrárny, pořízených termokamerou na dronu. Momentálně profesionál na inspekce elektráren manuálně analyzuje pořízené snímky a hledá v nich závady. Tento přístup je velmi časově náročný a zavedení určité míry automatizace by tento proces mohlo usnadnit a urychlit. S tímto záměrem jsem natrénoval a nasadil model neuronové sítě **U-Net**, který na snímcích detekuje nejčastější typ závady, kterým je „hot spot“. Pro vizualizaci nalezených závad jsem navrhl a vytvořil webovou aplikaci, která je zobrazuje v kompletní fotomozaice fotovoltaické elektrárny. V rámci aplikace může uživatel přidávat a mazat jednotlivé vady. Aplikace také umožňuje anotaci panelů, které jsou využity při tvorbě výsledné tabulky, jakmile je elektrárna kompletně anotována.

Abstract

The thesis describes the processing of thermal images of photovoltaic power plants captured by a drone. In contemporary solutions, the images are analyzed manually, where an expert in thermal imaging searches for defects in individual panels. This approach is very time-consuming, and introducing some level of automation could ease the process. Therefore, I trained and utilized a **U-Net** model that detects hot spots in the images. To visualize and present the defects to the user, I designed and created a web-based application that highlights them in a complete orthomosaic of the photovoltaic power plant. Within the application, a user can annotate PV panels in the power plant and manually remove, or add any defect. When the plant is wholly annotated, an export to a spreadsheet can be created, matching defects to the individual annotated panels.

Klíčová slova

fotovoltaické elektrárny, radiometrické snímky, detekce vad, klasifikace vad, neuronové sítě, strojové učení, dron, uživatelské rozhraní

Keywords

photovoltaic power plants, radiometric imaging, defect detection, defect classification, neural networks, machine learning, drones, user interface

Citace

HAUŽVIC, Filip. *Detekce a klasifikace vad panelů fotovoltaických elektráren z termokamery dronu.* Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Daniel Bambušek

Detekce a klasifikace vad panelů fotovoltaických elektráren z termokamery dronu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Daniela Bambuška. Další informace mi poskytl David Balhar. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

. . . . . .
Filip Haužvic
3. května 2024

Poděkování

Děkuji vedoucímu mé bakalářské práce, Ing. Danielovi Bambuškovi, za cenné rady a pomoc při vypracování mé závěrečné práce. Dále děkuji Davidovi Balharovi za poskytnutí znalostí z oblasti inspekcí elektráren a poskytnutí snímků pro trénovací datovou sadu.

Obsah

1 Úvod **4**

2 Shrnutí momentálního stavu **6**
2.1 Druhy závad fotovoltaických panelů 6
2.2 Průběh inspekce 7
2.3 Existující řešení 9

3 Přehled využívaných technologií **10**
3.1 Drony 10
3.2 Termokamery 13
3.3 Počítačové vidění a neuronové sítě 15

4 Návrh řešení **19**
4.1 Jednoduché metody počítačového vidění 19
4.2 Neuronová síť 21
4.3 Aplikace 23

5 Implementace **25**
5.1 Model pro detekci vad 25
5.2 Model pro detekci panelů 27
5.3 Aplikace 28

6 Testování a vyhodnocení **33**
6.1 Metriky 33
6.2 Modely pro detekci vad 35
6.3 Model pro detekci panelů 39

7 Závěr **41**

Literatura **42**

Seznam obrázků

2.1 Všechny druhy závad řešené v této práci. Fotografie jsou detailní záběry na jednotlivé vady, vystřižené z termografického snímku. Všechny vady se projevují rozdílnou vadou vůči svému okolí, různé druhy vad mají ovšem různé charakteristiky (tvar, umístění, intenzita). 7
2.2 Ručně anotovaný snímek v programu *Flir Tools*. Vady jsou ve snímku označeny jako body a seznam všech označených bodů (a teplot v těcho bodech) lze vidět na pravé straně. 8
2.3 Ukázka záznamu ve výsledné tabulce inspekce pro anotovaný snímek na obrázku 2.2. Do sloupce „označení vady“jsou použity názvy bodů v anotované fotografii (viz obrázek 2.2). Sloupce Část FVE, Řada, Lavice a Panel značí, na kterém panelu v elektrárně se panel nachází a slouží k jeho lokalizaci. . . 8

3.1 Možné konfigurace rámu dronu, převzato z [8]. 11
3.2 DJI Matrice 200 (zdroj: `https://www.dji.com/cz/matrice-200-series-v2`). 12
3.3 Elektromagnetické spektrum s rozděleným infračerveným spektrem, převzato z [3]. 13
3.4 Infračervená fotografie solárních panelů (zdroj: `https://thedronelifenj.com/benefits-of-infrared-drone-solar-inspections/`). 14
3.5 LWIR termokamera FLIR A6780 (zdroj: `https://www.flir.com/products/a6780-sls/`). 14
3.6 Perceptron. 16
3.7 „Multilayer Perceptron“(MLP). 16
3.8 Konvoluční vrstva s jedním filtrem (slidy dostupné v popisu videa `https://www.youtube.com/watch?v=ANyxBVxmdZ0`). 18

4.1 Odraz slunečního záření na solárním panelu (spodní část dolních panelů). . 19
4.2 Potenciální body hran (vlevo), nalezené pozadí (vpravo). 20
4.3 Původní fotografie (vlevo), výsledek extrakce panelů (vpravo). 21
4.4 Detekce teplotních špiček ve snímku. V horní části je samotná fotografie, ve které jsou nalezené vady označeny bílými křížky. Ve spodní části je graf obsahující teplotní hodnoty všech pixelů. Osa x značí pořadí pixelu (číslo řádku $\times$ počet prvků v řádku + pořadí v řádku) a osa y značí teplotu daného pixelu. 22
4.5 Výstup modelu *U-Net*. Fialová barva značí pozadí a žlutě jsou označeny nalezené závady typu hot spot. 23
4.6 Návrh hlavní anotační stránky 24

5.1 Anotovaný snímek použitý pro trénování Faster R-CNN. Růžové obdélníky jsou manuálně anotované vady typu hot spot. 26

5.2 Architektura modelu U-Net, převzato z [18]. 26
5.3 Anotovaný snímek pro trénování *U-Net* k sémantické segmentaci. Ručně vytvořené anotace značí vady typu hot spot (růžové) a nefunkční třetinu (modré). 27
5.4 Anotovaný snímek pro trénování *Mask R-CNN* k segmentaci instancí. Žluté obdélníky jsou manuálně vložené anotace značící panely ve fotografii. 28
5.5 Anotační stránka v aplikaci. Stránka zobrazuje fotomozaiku elektrárny s nalezenými vadami (nebo panely, dle vybraného režimu). Napravo je seznam vad, pod kterým jsou vypsané informace o uživatelem označené vadě. Označená vada je na fotomozaice zvýrazněna bílým kolečkem. Na levé straně fotomozaiky jsou tlačítka pro přepínámí režimu (zobrazení vad/panelů, výběr/tvorba). 29
5.6 Stránka pro detekci závad v jednotlivých snímcích. 30
5.7 Backend architektura. Červené části znázorňují koncové body. Modré části jsou zodpovědné za práci s daty a zelené části implementují logiku aplikace. 31
5.8 Koncové body aplikačního rozhraní. Zeleně koncové body pro anotace, zlutě koncové body pro panely, modře koncové body pro závady a červeně koncové body pro detekci. 32

6.1 Confusion Matrix. Řada určuje předpovězenou hodnotu (pozitivní, nebo negativní) a sloupec určuje očekávanou hodnotu. Umístění v matici určuje charakter předpovědi. Na diagonále jsou správné předpovědi (TP – „True Positive“, TN – „True Negative“), které odpovídají skutečnosti. Ostatní prvky matice (FN – „False Negative“, FP – „False Positive“) značí nesprávnou předpověď. 34
6.2 Precison-Recall křivka. 35
6.3 Graf z trénování *Faster R-CNN*. 36
6.4 *U-Net* Experiment 1. 37
6.5 *U-Net* Experiment 2. 37
6.6 *U-Net* Experiment 3. 38
6.7 *U-Net* Experiment 4. 38
6.8 Porovnání *Dice* skóre modelů při trénování. 39
6.9 Porovnání hodnot *Loss funkce* modelů při trénování. 39
6.10 Graf loss funkce z trénování Mask R-CNN pro detekci panelů. 40

Kapitola 1

Úvod

S narůstajícím počtem fotovoltaických elektráren po celém světě vznikají problémy týkající se údržby. U fotovoltaického panelu může dojít k závadám při výrobě, přepravě, instalaci i provozu. Pokud je chyba včas detekována, lze ji opravit a panel může nadále fungovat. Pokud je panel se závadou ponechán v provozu, dochází ke značným ztrátám generované energie, a také vzniká riziko požáru. Vady se totiž projevují jako oblast se zvýšenou teplotou. Pro nalezení těchto závad lze využít několik metod, více popsaných v sekci 2. Metodou podstatnou pro tuto práci je využití dronů vybavených termografickou kamerou. Pilot dronu naplánuje trasu, po které dron letí a pořizuje infračervené fotografie (termogramy) elektrárny. Tyto fotografie jsou poté ručně analyzovány. Na inspektorem vybraných fotografiích musí manuálně označit všechny závady a fotografii následně ukládá i s anotacemi. Při posledním kroku inspekce zapisuje inspektor označené vady do Excel tabulky. Výsledkem inspekce jsou tedy snímky s označenými vadami a vyplněná tabulka.

Podle tvaru a velikosti oblasti nalezené závady lze přiřadit do jedné z několika kategorií. Nejčastější závadou je jednotlivý bod se zvýšenou teplotou (hot spot). Ten se na termogramu projevuje jako světlý bod na fotovoltaickém panelu. Podle velikosti rozdílu teploty v tomto bodě oproti zdravé části panelu se dělí do tří kategorií. U vyšších teplotních rozdílů dochází k vyšším ztrátám. Hot spot je jediná závada, u které se pozoruje rozdíl teploty vůči zdravému bodu. Další závada, která se může objevit, se projevuje jako pás o vyšší teplotě pokrývající třetinu fotovoltaického panelu. Pokud se objeví mnoho vad typu „hot spot“ na jednom panelu, jedná se o prasklý modul. Zvýšenou teplotu může mít i celá řada panelů, což je vlastní typ závady. Poslední závada nastává, pokud okraj panelu překrývá zeleň. V tomto případě se zvýšená teplota projevuje na okraji panelu. Typy závad jsou detailněji popsány v sekci 2.1.

Proces manuální inspekce snímků pořízených dronem může trvat až 4 dny. Cílem této práce je vytvořit program, který dokáže vady detekovat a na snímcích označit. Samotná detekce závad je však pouze jedna část manuální inspekce snímků. Ve výsledku je v anotaci nutné zaznamenat, kde v elektrárně se nalezená vada nachází, čehož je docíleno zaznamenáním, na kterém panelu se závada nachází. Za tímto účelem je každý fotovoltaický panel označen několika charakteristikami (číslo lavice, číslo řady, část elektrárny). Vzhledem k tomu, že každý majitel elektrárny využívá jiné značení panelů, anotace panelů zatím nelze automatizovat. Mé řešení navrhuje způsob, který zjednoduší a urychlí anotaci jednotlivých panelů. Tyto anotace panelů lze poté využít při tvorbě výsledné tabulky. K umožnění anotace jednotlivých panelů je třeba tyto panely ve fotografii detekovat.

K detekci vad jsem navrhl detektor využívající konvoluční neuronovou síť **U-Net**. Detektor v této práci se zaměřuje na nejčastější typ závady – hot spot. Aplikační rozhraní

používané vytvořenou aplikací umožňuje snadnou výměnu modelu používaného k detekci. Pokud by tedy vytvořená datová sada byla rozšířena kvalitnějšími daty a byl by natrénován lepší model, který by detekoval více druhů vad, lze momentálně používaný model nahradit novým. Pro detekci panelů jsem využil existující implementaci modelu **Mask R-CNN**.

V kapitole 2 popisuji momentální procesy inspekce fotovoltaických elektráren, se zaměřením na postup dále diskutovaný v této práci. Tato kapitola také obsahuje popis existujících vědeckých i komerčních řešení úkolů dále řešených v této práci. Kapitola 3 obsahuje základní teoretické informace o tématech řešených v této práci. Kapitola 4 se zaměřuje na návrh výsledného řešení a kapitola 5 obsahuje informace o implementaci tohoto návrhu. V kapitole 6 prezentuji výsledky testování mého řešení. Kapitola 7 shrnuje výsledky mé práce.

Kapitola 2

Shrnutí momentálního stavu

Inspekce solárních panelů je důležitou součástí provozu fotovoltaické elektrárny. Různé typy závad mohou způsobovat značné ztráty generované energie a v nejzávažnějších případech může vznikat i riziko požáru. Pro inspekci fotovoltaických elektráren lze využít více přístupů. Nabízí se zde rozdělení podle toho, zda se při inspekci využívají kamery, nebo ne [11]. Během posledního desetiletí byl v odvětví automatizace inspekcí značný pokrok díky rostoucí popularitě dronů a neuronových sítí, každopádně odborník zatím kompletně nahradit nelze a stále je potřeba jeho zásah. Cílem snah o automatizaci tohoto procesu je momentálně minimalizovat čas potřebný pro inspekci.

2.1 Druhy závad fotovoltaických panelů

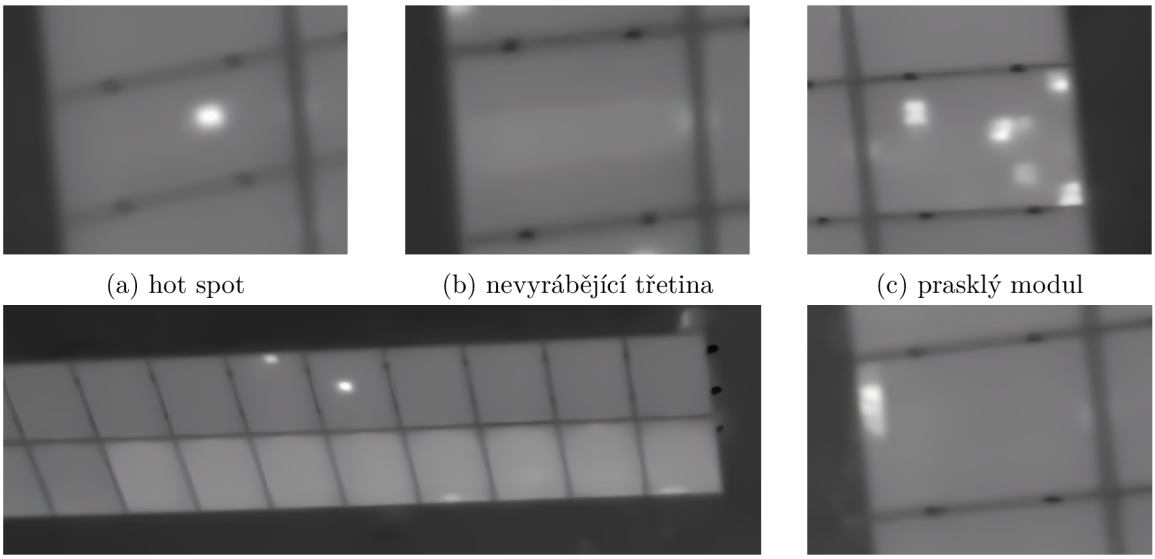
Vady fotovoltaických panelů se dají kategorizovat do široké řady typů, z nichž se některé vyskytují častěji než ostatní (může být ovlivněno například prostředím, ve kterém jsou solární panely instalovány). Různé metody inspekce jsou vhodné pro detekci určitých vad. Tuto práci jsem vypracoval ve spolupráci s profesionálem z firmy Arridere[1], provádějící inspekce. Vady, diskutované v této práci, jsem zvolil na základě toho, které typy závad hledal při inspekci doposud. Existuje i mnoho dalších vad, které v této práci nejsou blíže diskutovány, jako například delaminace, rezivění, nefunkční invertér, porucha akumulátorové baterie, apod [23].

Všechny vady řešené v této práci byly již stručně popsány v úvodu (viz kapitola 1) a na obrázku 2.1 lze vidět, jak se projevují na termálním snímku. Vady lze jednoznačně identifikovat podle jejich charakteristik na termálních snímcích (tvar, velikost). Nejčastější závadou je **hot spot** a méně častá je **nevyrábějící třetina**. **Nefunkční řada** a **prasklý modul** se nevyskytují zdaleka tak často, ovšem způsobují mnohem větší ztráty energie. **Zeleň** je nejméně závažnou vadou při inspekci, vzhledem k tomu, že okolí fotovoltaických panelů se (většinou) pravidelně seče, takže se tento problém vyřeší i bez inspekce. Prasklý modul společně s vadou typu hot spot nejvyšší kategorie mohou dokonce představovat nebezpečí požáru.

Hot spot lze poznat jako malý bod na fotovoltaickém panelu, nejčastěji kruhového tvaru. Dělí se do tří kategorií, podle velikosti rozdílu teploty bodu se zvýšenou teplotou a teploty nezávadného bodu v panelu. Za předpokladu, že:

$$x = \text{teplota bodu se zvýšenou teplotou} - \text{teplota zdravého bodu}$$

[1] http://www.arridere.cz/cs/



(a) hot spot (b) nevyrábějící třetina (c) prasklý modul

(d) nefunkční řada (e) zeleň

Obrázek 2.1: Všechny druhy závad řešené v této práci. Fotografie jsou detailní záběry na jednotlivé vady, vystřižené z termografického snímku. Všechny vady se projevují rozdílnou vadou vůči svému okolí, různé druhy vad mají ovšem různé charakteristiky (tvar, umístění, intenzita).

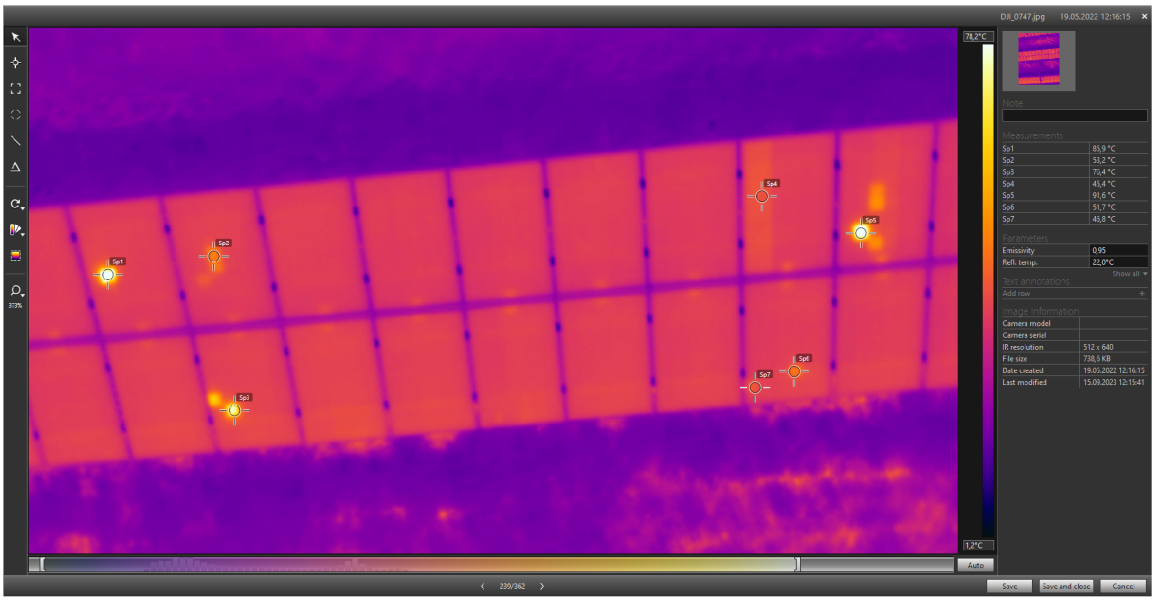
platí:

- kategorie 1 – $10°C < x < 15°C$
- kategorie 2 – $15°C < x < 30°C$
- kategorie 3 – $30°C < x$

Jeden hot spot kategorie 3 může způsobit ztrátu až 20 % generované energie daným panelem. U závady typu nevyrábějící třetina se teplotní rozdíl oproti zdravému panelu neřeší, detekuje se pouze podle pásu se zvýšenou teplotou pokrývající zhruba třetinu panelu. Nevyrábějící třetina způsobuje ztrátu 33 % generované energie. Nefunkční řada se projevuje jako dva a více panelů vedle sebe se zvýšenou teplotou, což může způsobovat vysoké ztráty, pokud je těchto vadných panelů v řadě mnoho. Prasklý modul lze poznat podle velkého množství závad typu hot spot vedle sebe, což způsobuje že panel generuje velmi málo energie a v nejhorším případě tato závada vytváří riziko požáru. Prasklý modul a nefunkční řada jsou typy závad, které nejsou příliš časté. Vada typu zeleň nelze poznat pouze z termálního snímku. Nejčastěji se vyskytuje na okraji panelu, není to ovšem nutnou podmínkou. Je nutné tuto vadu ověřit s barevnou fotografií stejného panelu, ale ani tento krok nemusí identifikovat tuto vadu naprosto jednoznačně. Zeleň je poměrně snadné zaměnit za hot spot. Jak již bylo zmíněno, závada typu zeleň není během inspekce podstatnou, pokud je elektrárna pravidelně udržována a okolní zeleň sečena.

2.2 Průběh inspekce

Jak jsem již zmínil, existuje více přístupů pro inspekci fotovoltaických elektráren. Jedním z přístupů je využití senzorů elektrického proudu a napětí, rozmístěných na různých místech



Obrázek 2.2: Ručně anotovaný snímek v programu *Flir Tools*. Vady jsou ve snímku označeny jako body a seznam všech označených bodů (a teplot v těcho bodech) lze vidět na pravé straně.

ID vady	ID termo snímku	Označení vady	Část FVE	Řada (J->S)	Lavice (Z->V)	Panel (L->P, H/D)	Druh poškození	Ztráta	Navrhované řešení
206	DJI_0747	Sp1	A1	14	2	5H	Hot spot kat. 3	Až 20%	Výměna
207		Sp2	A1	14	2	6H	Hot spot kat. 1	Nízká	Sledování
208		Sp3	A1	14	2	6D	Hot spot kat. 3	Až 20%	Výměna
209		Sp4	A1	14	2	18H	Nevyrábějící 1/3	33%	Výměna/oprava
210		Sp5	A1	14	2	19H	Hot spot kat. 3	Až 20%	Výměna
211		Sp6	A1	14	2	18D	Hot spot kat. 1	Nízká	Sledování
212		Sp7	A1	14	2	19D	Hot spot kat. 1	Nízká	Sledování

Obrázek 2.3: Ukázka záznamu ve výsledné tabulce inspekce pro anotovaný snímek na obrázku 2.2. Do sloupce „označení vady“ jsou použity názvy bodů v anotované fotografii (viz obrázek 2.2). Sloupce Část FVE, Řada, Lavice a Panel značí, na kterém panelu v elektrárně se panel nachází a slouží k jeho lokalizaci.

fotovoltaické elektrárny. Tento proces inspekce není pro tuto práci podstatný a více se o něm lze dočíst v [11]. U inspekcí využívajících kamer převažují tři druhy snímků: snímek viditelného světla, termální snímek a elektroluminescenční snímek. Každý druh snímku umožňuje detekci jiných typů závad, jelikož zachycují různé vlastnosti. Snímky viditelného světla lze pořídit běžnými kamerami. Pro získání termálních snímků je třeba speciálních termokamer, blíže popsaných v sekci 3.2. Elektroluminescenční kamery zachycují záření vznikající při průchodu elektrického proudu vhodným materiálem (luminoformem).

Přístup, který se tato práce snaží zjednodušit, využívá termálních snímků. Ty jsou pořízeny termokamerou připevněnou k dronu. Pilot dronu naplánuje trasu letu v areálu elektrárny tak, aby dron zachytil všechny panely v elektrárně. Následně tyto fotografie manuálně prochází expert v oblasti termální fotografie, a označuje chyby ve snímcích nástrojem *Flir Tools*[2] (viz obrázek 2.2). Typy hledaných závad jsou popsány v sekci 2.1. Každá nale-

[2] https://support.flir.com/SwDownload/app/RssSWDownload.aspx?ID=1247

zená závada musí být ručně vložena do Excel tabulky (viz obrázek 2.3) a je třeba u ní uvést i její pozici v celé elektrárně. K tomu se využívá označení panelu, na kterém se vada nachází. Označení panelu se většinou skládá z části elektrárny, čísla řady, čísla lavice, a pozice panelu v lavici. Jednotlivé panely tvoří lavice, které jsou nejčastěji postaveny v řadách. Neexistuje ovšem jednotný systém pro označování panelů a každý majitel fotovoltaické elektrárny využívá vlastní, kvůli čemuž zatím proces nelze automatizovat. Tabulka obsahující všechny vady je společně s anotovanými snímky finálním produktem inspekce.

2.3 Existující řešení

Výzkum automatické detekce vad na fotovoltaických panelech v posledních deseti letech rapidně narůstá na popularitě [10], díky rostoucímu podílu využití obnovitelných zdrojů a solární energie, a zejména kvůli objevům v oblasti neuronových sítí. I přes popularitu neuronových sítí existují práce využívající klasické metody počítačového vidění (detekce hran [7], K-means [19]) k automatické detekci vad. Rozhodně ale převažují metody využívající neuronové sítě [10] a k detekci vad z fotografií zejména konvoluční neuronové sítě, které jsou blíže popsány v sekci 3.3. Při výzkumu automatické detekce vad z fotografií jsou častěji než termální snímky využívány elektroluminescenční snímky [10]. Dle mého průzkumu, práce, které zpracovávají termální fotografie, využily následující architektury neuronových sítí: *VGG-16* [15], *MobileNet* [2], *(Faster) R-CNN* [24, 6] a také vlastní navrženou architekturu [1, 12]. Tyto práce detekují buď vady typu hot spot, nebo klasifikují jestli je daný panel nefunkční. Detekce různých druhů závad nebyla součástí dočasných prací zpracovávajících termální snímky.

Nalezl jsem i dvě existující komerční aplikace[3,4], které pomáhají při inspekci s využitím termálních fotografií a obsahují i možnost automatické detekce závad. Ze stránek těchto programů však není jasné, s jakou přesností dokáží detekovat různé defekty. Při využití těchto řešení je také nutno vzít v potaz, jestli se využití vyplatí s ohledem na jejich cenu. Profesionál, se kterým jsem tuto práci konzultoval, nevyužíval žádný momentálně dostupný komerční software pro provádění inspekcí.

Součástí mnou navrženého řešení je i detekce fotovoltaických panelů ve fotografiích. Jedná se o relativně snadný úkol, jelikož se v podstatě jedná o detekci obdélníkových objektů ve fotografii. Na takový úkol lze využít již existující řešení (např. knihovna *Detectron2*[5]), protože problém detekce objektů je jeden ze základních úkolů počítačového vidění [25]. Existují i práce, které řeší přímo úkol detekce panelů za pomocí *Mask R-CNN* [22], *U-Net* [14], nebo *Faster R-CNN* [4].

[3]Scopito: https://scopito.com/solar-pv-inspection-software/
[4]Raptor Maps: https://raptormaps.com/products/solar-aerial-inspections
[5]https://github.com/facebookresearch/detectron2

Kapitola 3

Přehled využívaných technologií

V této kapitole shrnuji informace, které považuji za podstatné pro pochopení řešeného problému a mnou navrženého řešení. Technologie diskutované v této kapitole jsou drony (viz sekce 3.1), termokamery (viz sekce 3.2) a počítačové vidění (viz sekce 3.3).

3.1 Drony

Drony jsou bezpilotní letouny. V podstatě se jedná o roboty, kteří jsou ovládány buď dálkově prostřednictvím dálkového ovládání vysílající rádiové vlny, nebo autonomně, kdy následují předem danou trasu.

Historie

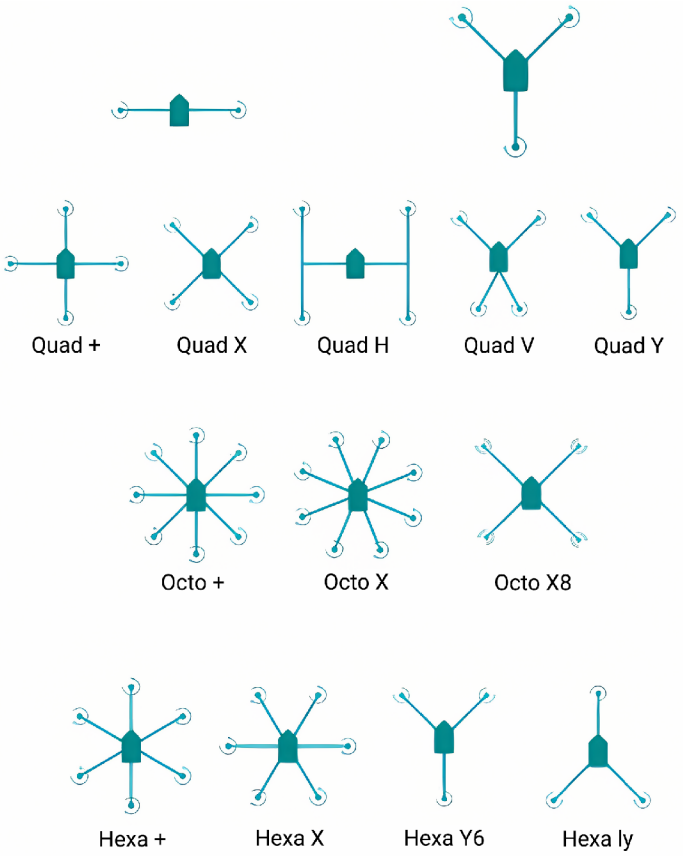
Historie dronů sahá téměř dvě století zpět. První pokusy o jejich využití se objevily již v roce 1849 během První italské války o nezávislost. První drony měly podobu horkovzdušného balónu naplněného výbušninami. První dálkově ovládaný dron byl vyvinut podle návrhu Nikoly Tesly v roce 1898. Bezpilotní letouny byly původně využívané exkluzivně pro vojenské účely. Mezi významné milníky patří sériová výroba bezpilotního letounu OQ-2 v roce 1939 a vývoj bezpilotních letounů schopných sledování a sbírání informací v období studené války [13].

Ve 21. století přešly drony od primárně vojenských aplikací k širokému komerčnímu a civilnímu využití. Společnosti jako DJI Technology a Parrot se staly průkopníky dronů pro spotřebitele. Inovativní iniciativy, jako je program doručování balíků drony společnosti Amazon a projekt Aquila společnosti Facebook, ukázaly rozmanitý potenciál dronů mimo vojenský kontext [13]. V současné době hrají bezpilotní letouny zásadní roli v různých odvětvích, jako například inspekce infrastruktur, zemědělství, nebo u složek integrovaného záchranného systému.

Komponenty

Základní částí dronu je jeho rám. Rám by měl být co nejlehčí. Konstrukce rámu je určena počtem ramen, které drží jednotlivé motory. Obecně lze říct, že vyšší počet ramen vede ke stabilnějšímu letu. Možné konfigurace rámu lze vidět na obrázku 3.1.

Další důležitou částí dronu jsou samotné motory a vrtule. Tyto komponenty tvoří hlavní pohonný systém dronu a jsou vystaveny největšímu zatížení, proto je velmi důležitá jejich životnost. Umístění vrtulí vzhledem ke směru pohybu je důležitou vlastností. Vrtule, která je



Obrázek 3.1: Možné konfigurace rámu dronu, převzato z [8].

vepředu vzhledem k pohybu dronu se nazývá vedoucí vrtule. Nejběžnější rozložení vrtulí je takzvané rozložení „X“, kde jsou dvě vedoucí vrtule a celkový počet vrtulí je sudý. Všechna možná rozložení vrtulí lze vidět na obrázku 3.1. V každém rozložení lze použít dvojitou vrtuli, což znamená že je vrtule jak nahoře, tak i dole. Dvojitá vrtule značně zvyšuje sílu dronu bez potřeby přidání dalšího ramene. Samotné křídla vrtule jsou vyrobena z uhlíkových vláken, plastu, nebo hliníku [8].

Drony jsou závislé na bateriích, které obvykle vydrží přibližně 20-50 minut[1] letu. Chemické reakce v bateriích přeměňují chemickou energii na elektrickou. Existují různé typy baterií, včetně zinko-uhlíkových, stříbro-zinkových a lithiových. Drony často používají lithium-polymerové (Li-Po) baterie, které fungují podobně jako lithium-iontové články (Li-ion). Tyto baterie umožňují mnoho nabíjecích cyklů. Navíc se lithium-iontové akumulátory vybíjejí pomaleji než akumulátory využívající nikl a mají vyšší výstupní napětí. Akumulátory je nutno často dobíjet a skladovat v chladném prostředí, aby se zachovala jejich životnost. Li-ion články mají široký rozsah provozních teplot a vykazují minimální samovybíjení. Rozdíly mezi články Li-ion a Li-Po zahrnují jmenovité napětí, maximální napětí a odolnost proti mechanickému poškození. Požadavky na elektrochemické akumulátory zahrnují malé rozměry, nízkou hmotnost, rychlé nabíjení, vysokou bezpečnost, odolnost, minimální samovybíjení a cenovou dostupnost [8].

[1] https://soldrones.com/blog/drone-battery-life/

Obrázek 3.2: DJI Matrice 200 (zdroj: https://www.dji.com/cz/matrice-200-series-v2).

Drony jsou ovládány elektronickým řídícím a komunikačním systémem. Ten dohlíží na řízení letu, stabilitu a reakci na vnější síly prostřednictvím různých komponentů. Mezi hlavní komponenty řídícího systému patří letový regulátor, elektronická kontrola rychlosti, napájecí jednotka, telemetrický modul, kamera s detekcí vzdálenosti a řídicí jednotky motorů. Tyto součásti zajišťují optimální výkon a bezpečnost řízením otáček motoru, napájení, přenosu telemetrických dat, opatření proti kolizi a monitorování baterie. Programování a kalibraci zjednodušuje programovací karta, která dokáže měnit parametry dronu. Komunikace s drony probíhá prostřednictvím Wi-fi připojení k síti, rádiových vln, nebo Bluetooth. Řídicí algoritmy využívají senzory, jako jsou gyroskopy a akcelerometry, k udržení stability a polohy za letu [8].

Využití dronů

Z historie vývoje dronů je zřejmé, že hlavním důvodem pro jejich vývoj bylo vojenské využití (viz sekce 3.1). Ve 21. století se ale drony začínají využívat i ve spoustě jiných oblastí. V této sekci je stručně popsáno, v jakých odvětvích dnes drony nalezneme. Detailnější popis využití v této práci lze nalézt v kapitole 2.

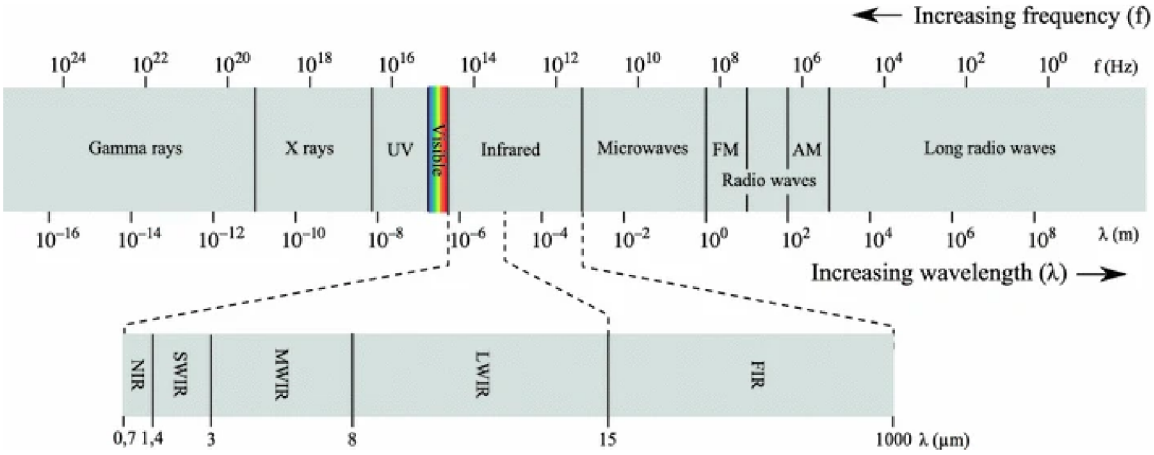
V zemědělství se drony využívají pro sběr dat a pozorování plodin, nebo půdy. Mohou ale také být využiti pro zavlažování či hnojení, nebo také sázení. V oblasti ochrany životního prostředí využití zahrnuje sledování ohrožených druhů, pomoc při obnově lesů, nebo monitorování úniků ropy a plynu. V případě mimořádných událostí pomáhají drony pracovníkům první pomoci zajištěním průzkumu terénu, navádění záchranných týmů a pomoci při pátracích a záchranných misích. Fotografický a filmový průmysl využívá drony pro pořizování leteckých záběrů ve filmech, televizních pořadech a reklamách [21]. Drony se také využívají při inspekcích infrastruktur, jako třeba budov, mostů, nebo větrných a fotovoltaických elektráren. Pro pořízení snímků fotovoltaických panelů, použitých v této práci, byl využitý dron *DJI Matrice 200*, zobrazený na obrázku 3.2.

3.2 Termokamery

Závady na fotovoltaických panelech se projevují jako tepelné anomálie. Běžné kamery zachycují elektromagnetické záření ve viditelném spektru a tvoří barevné fotografie. Z těchto fotografií není možné rozpoznat výskyt tepelných rozdílů. Proto se pro inspekci fotovoltaických elektráren využívají jiné druhy kamer, jedním z nichž jsou termokamery. Jak termokamera může vypadat lze vidět na obrázku 3.5.

Infračervené záření

Každý objekt s teplotou vyšší než absolutní nula vyzařuje infračervené záření. Infračervené záření se na elektromagnetickém spektru nachází mezi viditelným zářením a mikrovlnami. Vlnová délka infračerveného záření je v rozmezí $0.7 - 1000\mu m$. Infračervené spektrum lze rozdělit do několika regionů. Infračervené záření se střední vlnovou délkou (mid-wavelength infrared, MWIR) a dlouhou vlnovou délkou (long-wavelength infrared, LWIR) bývají označovány jako termální záření. Objekty o teplotách $190 - 1000K$ vyzařují infračernevé záření právě s těmito vlnovými délkami (viz obrázek 3.3) [3].



Obrázek 3.3: Elektromagnetické spektrum s rozděleným infračerveným spektrem, převzato z [3].

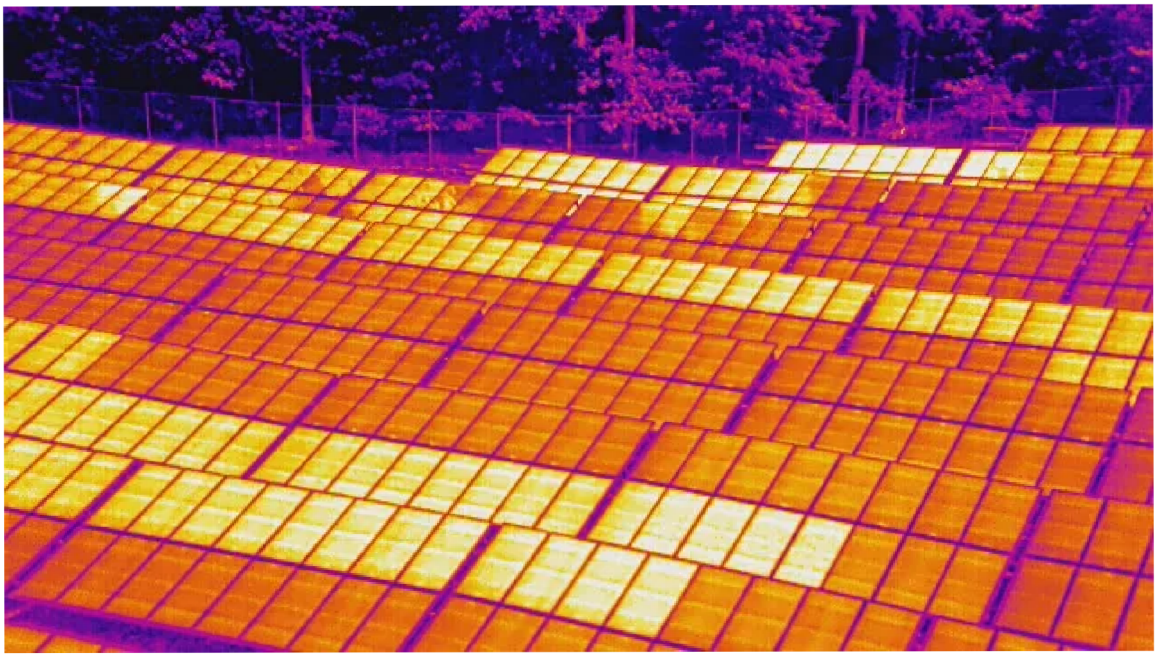
Princip fungování

Termokamery využívají detektory, které snímají jeden bod, nebo řadu výsledné fotografie. Tyto detektory měří intenzitu infračerveného záření v daném bodě. Pokud jsou tyto detektory uspořádány do dvojrozměrného pole, jedná se o „infrared focal plane array (IRFPA)“. Fotografie je pořízena tak, že všechny části fotografie jsou zaznamenány v jeden okamžik.

Detektory v termálních kamerách mohou být dvojího typu: fotonové, nebo termální. Fotonové detektory přímo konvertují absorbované elektromagnetické záření na elektrickou energii. Termální detektory převádí absorbované elektromagnetické záření na tepelnou energii. Nárůst teploty způsobí změnu fyzické vlastnosti materiálu, například odpor závislý na teplotě. Fotonové detektory běžně pracují s MWIR zářením. Oproti termálním detektorům umožňují vyšší snímkovou frekvenci. Hlavní nevýhodou fotonových detektorů je nutnost chlazení na teplotu nižší než $77K$, z důvodu snížení tepelného šumu. Lze tedy říci, že fotonové detektory poskytují přesnější a kvalitnější obraz, za cenu náročnější údržby a vyšší

ceny. Termální detektory pracují s LWIR zářením a pro převod tepelné energie na elektrickou využívají převážně mikrobolometry.

Výstupem typicky bývá fotografie stupně šedi, s přesností 8 až 16 bitů na každý pixel. Fotografie často bývají zobrazovány v různých barevných paletách, pro jednodušší lidskou interpretaci [3]. Výslednou fotografii pořízenou termokamerou lze vidět na obrázku 3.4.



Obrázek 3.4: Infračervená fotografie solárních panelů (zdroj: https://thedronelifenj.com/benefits-of-infrared-drone-solar-inspections/).

Obrázek 3.5: LWIR termokamera FLIR A6780 (zdroj: https://www.flir.com/products/a6780-sls/).

Parametry

Mezi důležité parametry při výběru termokamery patří teplotní rozsah, spektrální rozsah, teplotní citlivost, rozlišení, zorné pole a také cena. Teplotní rozsah určuje nejnižší a nejvyšší

teplotu, kterou dokáže kamera změřit. Spektrální rozsah určuje rozsah vlnových délek, které je schopná kamera detekovat (viz obrázek 3.3). Kamery s různým spektrálním rozsahem se často používají pro různé účely. LWIR kamery se používají pro „nízké“ teploty v rozsahu $-50°C - 900°C$, NIR (near infrared) pro vysoké teploty v rozsahu $450°C - 2450°C$ a MWIR pro nízké a střední teploty s vysokou přesností, ale i vysokými náklady. Zmíněné kamery lze využít i pro jiné účely, ale je nutné počítat se sníženou přesností. Tepelná citlivost (Noise Equivalent Temperature Difference, NETD) kamery určuje nejmenší rozdíl teplot, který kamera zvládne změřit. Požadavky na citlivost závisí na cílovém využití, ale obecně platí, že čím nižší je teplotní citlivost kamery, tím je výsledná fotografie přesnější. Rozlišení termokamery určuje rozlišení výstupního snímku. Rozlišení také v podstatě určuje počet detektorů, které se v kameře nachází. Komerční termokamery nejčastěji používají rozlišení v rozmezí $80x80 - 640x480$ pixelů. Zorné pole je dáno použitým objektivem. Čím větší je zorné pole, tím větší prostor dokáže kamera zachytit. Některé kamery umožňují vyměnit objektiv[2]. Vzhledem k nízké propustnosti infračerveného záření u skla se pro objektiv používají jiné materiály. Nejčastěji je objektiv vyráběn z germania, které má poměrně vysokou cenu [3].

3.3 Počítačové vidění a neuronové sítě

Počítače jsou v mnoha úkolech neporovnatelně lepší než lidé. Pokud se jedná například o zpracování velkého množství dat, nebo provádění komplexních výpočtů, počítače dokážou tyto úkoly provádět s rychlostí, přesností a konzistencí, které se člověk nemůže rovnat. Stále ale existují aktivity, ve kterých počítače zaostávají za lidmi. Většinou se jedná o aktivity vyžadující kreativitu, emoční cítění, nebo intuici. Právě intuice a zkušenosti umožňují lidem zjistit z fotografie mnoho informací. S lehkostí zvládneme rozlišit tvary a objekty ve fotografii, a určit co reprezentují. Počítačové systémy s tímto úkolem ale mají značný problém, a pro úkol pro člověka velmi jednoduchý již několik desítek let vzniká mnoho technik a algoritmů [20], snažící se napodobit schopnost člověka získat z fotografie sémantické informace. Tyto snahy spadají pod odvětví nazývané počítačové vidění („Computer Vision“).

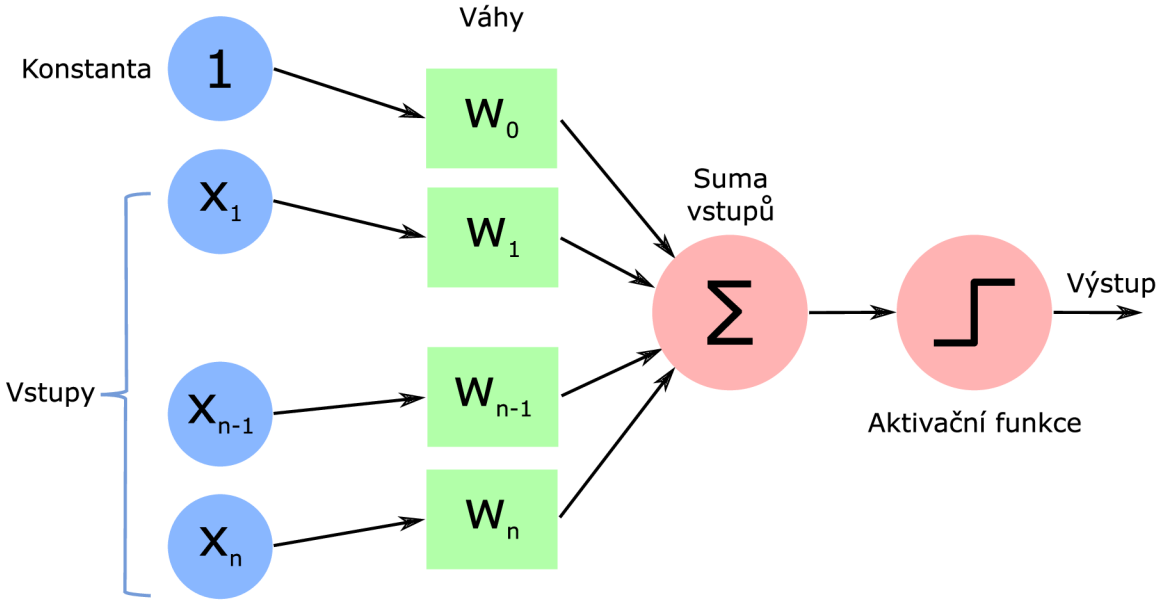
Počítačové vidění často využívá metod z odvětví strojového učení (anglicky „Machine Learning“). Strojové učení mimo jiné může využívat i neuronové sítě, které přibližně v posledním desetiletí získaly velkou popularitu. Konvoluční neuronové sítě se v počítačovém vidění staly velmi rozšířenými, díky jejich schopnostem získat z fotografie sémantické informace a následně s nimi pracovat. Cílem této práce není kompletně vysvětlit fungování neuronových sítí, ale stručné vysvětlení principu jejich fungování a procesu trénování. Tyto znalosti se mohou čtenáři hodit v pozdějších částech této práce.

Struktura neuronových sítí

Základní jednotkou neuronových sítí je jeden neuron, pro který se používá název perceptron. Perceptron má vektor vstupů o velikosti n, vektor vah o stejné velikosti n a jednu výstupní hodnotu. Všechny vstupní hodnoty vynásobené příslušnou váhou se následně sečtou a výsledná suma je vstupem pro aktivační funkci. Výsledek aktivační funkce je finálním výstupem perceptronu (viz obrázek 3.6)[3].

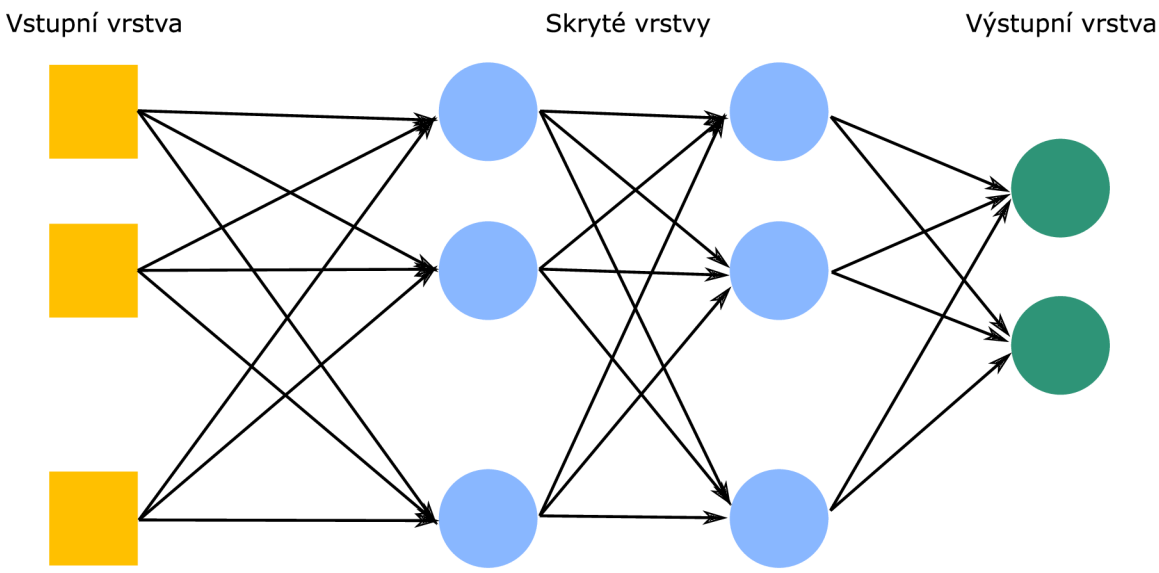
[2] https://visiontir.com/7-factors-when-choosing-a-thermal-camera-for-temperature-measurement/

[3] https://deepai.org/machine-learning-glossary-and-terms/perceptron



Obrázek 3.6: Perceptron.

Perceptron dokáže klasifikovat lineárně rozdělitelná data. Pro klasifikaci dat, které nelze lineárně rozdělit lze perceptrony spojit mezi sebou. Například výstup jednoho perceptronu může být vstupem jiného. Takto lze perceptrony dělit do vrstev, kde vstupy jsou hodnoty z předchozích vrstev a výstupy jsou zároveň vstupy pro následující vrstvu. Taková struktura se nazývá neuronová síť. Model neuronové sítě který vznikne propojením všech perceptronů z předchozí vrstvy na vstup každého perceptronu v aktuální vrstvě se nazývá „Multilayer Perceptron“ (MLP)[4] (viz obrázek 3.7).



Obrázek 3.7: „Multilayer Perceptron“ (MLP).

[4] https://deepai.org/machine-learning-glossary-and-terms/multilayer-perceptron

Vstupy do MLP tvoří první vrstvu a každý vstup je přiveden na vstup každého perceptronu v následující vrstvě. Vstupy mohou tvořit například hodnoty intenzity pixelu ve fotografii, ale obecně se může jednat o libovolné číselné hodnoty. Ve výstupní (poslední) vrstvě se nachází n perceptronů, kde n je počet tříd, do kterých chceme vstup klasifikovat. Výstup každého perceptronu v poslední vrstvě udává pravděpodobnost, že vstup patří do dané třídy. Vrstvy mezi vstupní a výstupní vrstvou se nazývají skryté vrstvy (anglicky „Hidden layers“).

Trénování neuronové sítě

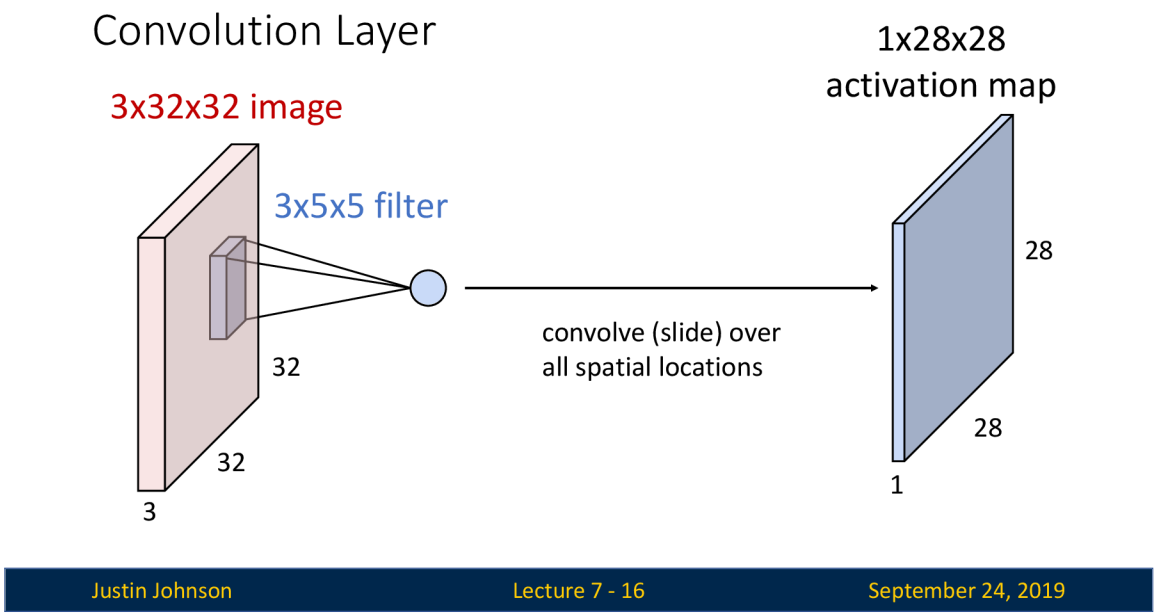
Správnost výsledků vyhodnocených neuronovou sítí závisí na parametrech jednotlivých prvků modelu. V případě MLP se jedná o váhy všech perceptronů. Používané modely mohou mít miliony parametrů a není možné je nastavit ručně. Využívá se proto proces trénování modelu. Trénování probíhá v několika krocích a běžně je třeba mít připraveno velké množství anotovaných dat (tzv. datová sada). Prvně model vyhodnotí vstup z datové sady. Následně se vypočítá vzdálenost odhadu modelu od předem určených správných hodnot, neboli velikost chyby odhadu. K výpočtu velikosti chyby se používají různé účelové funkce (anglicky „Loss function“). Nakonec se s využitím algoritmu zpětného šíření chyby[5,6] („Backpropagation“) upraví hodnoty parametrů tak, aby snížily hodnotu účelové funkce s momentálním vstupem. Tento proces se opakuje pro všechny anotované vstupy v datové sadě. Jeden průchod datovou sadou při trénování se označuje jako epocha. Znalost účelových funkcí nebo algoritmu zpětného šíření chyby není pro pochopení této práce třeba detailně chápat.

V této práci se také často využívá pojem „Transfer learning“. Jedná se o postup, při kterém se využije model neuronové sítě (většinou) natrénovaný na velké datové sadě. U konvoluční neuronové sítě se takto natrénovaný model naučil dobře rozpoznat různé charakteristiky ve fotografii. Tento model je poté trénován na vlastní datové sadě, aby se naučil mapovat charakteristiky na třídy, které chceme rozpoznávat. Touto metodou lze dosáhnout značně lepších výsledků při použití menší (<1000 anotací) datové sady a zkrátit dobu potřebnou pro trénování.

Konvoluční vrstva

Vrstvy neuronové sítě nemusí být tvořeny pouze perceptrony. Pro práci s obrazem se velmi často používají konvoluční vrstvy. Konvoluční vrstvy pracují s filtry. Předpokládejme, že vstupní obraz má rozměry $32 \times 32 \times 3$, kde 32 jsou šířka a výška obrazu a 3 je počet vstupních kanálů. Velikost konvolučního filtru lze určit, pro tento příklad předpokládejme rozměry $5 \times 5 \times 3$. Jedno důležité omezení je, že filtr musí ve standardní konvoluční vrstvě zpracovávat všechny kanály najednou, tedy v tomto případě musí poslední rozměr mít velikost 3. Filtr je pokládán na každý bod vstupního obrazu a výsledná hodnota v bodě je výsledkem konvoluce filtru a obrazu. Výstupem v tomto příkladu by tedy byl útvar o rozměrech $28 \times 28 \times 1$ (viz obrázek 3.8). Rozměry se zmenšily kvůli tomu, že v okrajích vstupní fotografie nelze provést konvoluce, vzhledem k nedefinovaným hodnotám mimo obraz, kam by filtr zasahoval. Pokud by zmenšení rozměrů nebylo žádoucí, lze například vyplnit prostor okolo obrazu hodnotami 0, nebo rozšířit hodnotu v okraji obrazu.

[5] https://www.3blue1brown.com/lessons/backpropagation
[6] https://www.youtube.com/watch?v=dB-u77Y5a6A



Obrázek 3.8: Konvoluční vrstva s jedním filtrem (slidy dostupné v popisu videa https://www.youtube.com/watch?v=ANyxBVxmdZ0).

Jedna konvoluční vrstva pracuje s několika takovými filtry, a proto mají výstupy konvoluční vrstvy rozměry $W \times H \times C$, kde W, H jsou šířka a výška obrazu a C je počet filtrů v konvoluční vrstvě. Výstupem filtru jsou různé vlastnosti předchozího obrazu. Jeden filtr tedy může například detekovat vodorovné hrany ve fotografii, a výstupem bude mapa zobrazující body ve fotografii, kde se takové hrany nachází. Výstupem konvoluční vrstvy je tedy několik map, každá vyjadřující nějakou vlastnost ze vstupního obrazu. Konvoluční vrstvy je možné skládat i několikrát za sebe. V takovém případě pozdější vrstvy nepracují s původním vstupním obrazem, ale mapami vlastností z předchozí konvoluční vrstvy. Hodnoty bodů ve filtrech jsou naučitelným parametrem, který se upravuje během trénování.

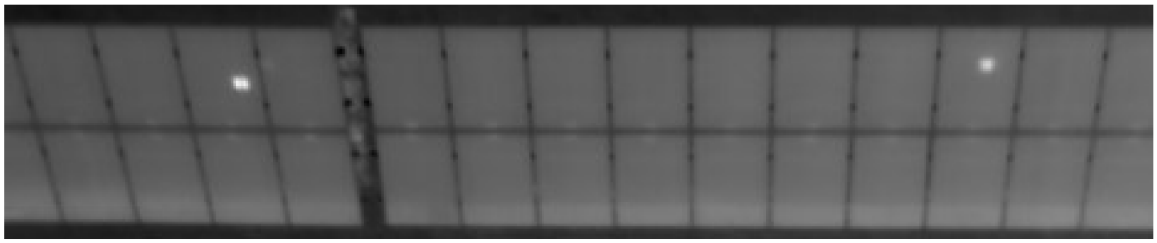
Kapitola 4

Návrh řešení

Pro detekci závad jsem zprvu zkoušel jednoduché metody počítačového vidění (popsané v sekci 4.1), které se ukázaly být nedostatečné, tak jsem se rozhodl využít neuronovou síť. Provedl jsem několik experimentů (popsaných v kapitole 6) s modely neuronové sítě pro detekci objektů a sémantickou segmentaci (blíže popsané v sekci 4.2). Výsledkem je detektor závad postavený na architektuře **U-Net**. Pro prezentaci výsledků jsem následně vytvořil aplikaci, jejíž návrh je popsán v sekci 4.3.

4.1 Jednoduché metody počítačového vidění

Jako první jsem vyzkoušel jednoduché prahování („Thresholding“), tedy vytvoření masky ze všech pixelů s intenzitou nad určitou prahovou hodnotou. Následně jednotlivé útvary v této masce byly považovány za vady. Tento přístup se potýkal hned s několika potížemi. Prvním problémem byly odrazy slunečního záření na určitých místech panelů. Tyto odrazy se na snímcích projevují jako světlejší regiony na solárních panelech. Odraz může být téměř nerozpoznatelný od jiných typů závad, pokud se díváme pouze na jedinou fotografii. Fakt, že se jedná o odrazy lze vypozorovat až při zobrazení fotografií z blízkého okolí, na kterých je patrné, že se tyto oblasti mezi snímky pohybují. Druhým problémem při prahování byly



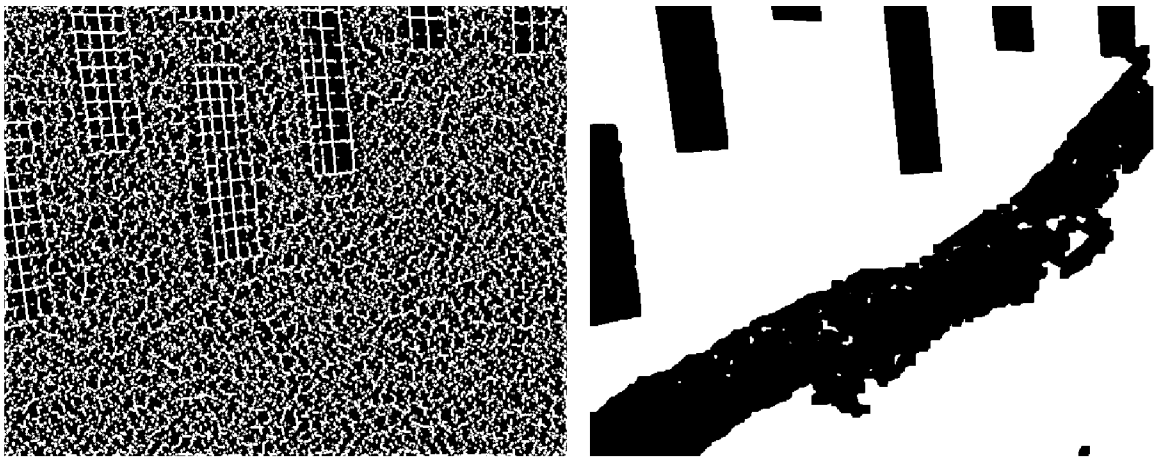
Obrázek 4.1: Odraz slunečního záření na solárním panelu (spodní část dolních panelů).

body s vysokou teplotou, které se nacházely mimo solární panely. Tento přístup nedokáže rozpoznat, jestli je závada na solárním panelu, nebo mimo něj.

Pro vyřešení problému teplotních anomálií v prostoru jsem zvolil přístup rozpoznání jednotlivých panelů ve fotografii. Pro řešení tohoto úkolu jsem následoval postup ze článku [9]. Jako první jsem nalezl potenciální hrany panelů (viz obrázek 4.2) následujícím postupem:

1. Konvoluce vstupního snímku s horizontálním a vertikálním filtrem o rozměrech 5×5 (Každá konvoluce je provedena na původní fotografii, výsledkem jsou 2 fotografie).

2. Výsledek konvoluce s vertikálním filtrem je zpracováván po řádcích a výsledek konvoluce s horizontálním filtrem je zpracováván po sloupcích.
3. Je uložena nejvyšší hodnota pro daný řádek, nebo sloupec. Okolí této hodnoty se vynuluje.
4. Předchozí krok se opakuje, dokud je v daném řádku, nebo sloupci, nenulová hodnota.
5. Všechny body ze všech sloupců a řádků se zkombinují do jednoho snímku.
6. Je provedena konvoluce výsledného snímku s kruhovým filtrem.



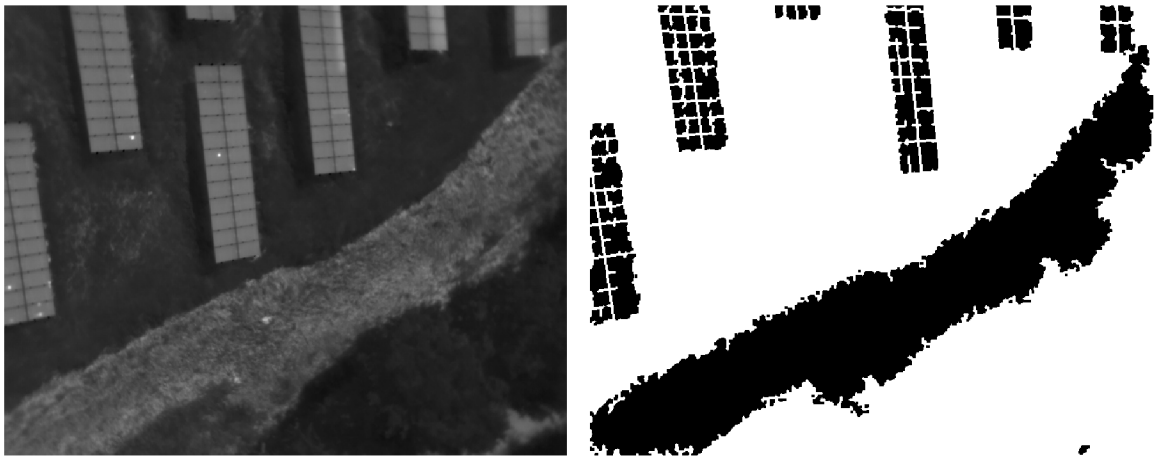
Obrázek 4.2: Potenciální body hran (vlevo), nalezené pozadí (vpravo).

Další krok v tomto procesu je odstranění pozadí ve fotografii. Pozadím se myslí oblasti, ve kterých se nenachází žádný solární panel.

1. Prahování. Zde je jedna z nevýhod této metody, zvolený práh značně ovlivňuje výsledek. Zvolit ale jeden univerzální práh pro obecná data je téměř nemožné).
2. Vyplnění děr. Díra je oblast, která není propojena s okrajem fotografie po prahování.
3. Dilatace (jádro o velikosti 3×3).
4. Vyplnění děr.
5. Eroze.

Tento proces eliminace pozadí nedokáže kompletně izolovat jednotlivé panely. Z tohoto důvodu se zkombinují potenciální body hran s nalezeným pozadím. Článek, který tento postup navrhl dále provádí úpravu tohoto snímku, který vznikne po kombinaci. Už v tomto bodě však bylo zjevné, že tento postup pro má vstupní data není vhodný. Jeden problém nastává při druhém kroku odstraňování pozadí, kdy proběhne vyplnění děr v pozadí. Solární panely v původním článku byly vždy v takové formaci, kde se dotýkaly okraje fotografie. V mém případě tento předpoklad ale nemusí odpovídat realitě, a tedy vyplnění děr v určitých snímcích odstraní i celou lavici fotovoltaických panelů (jak lze vidět na obrázku 4.3).

Druhý přístup byl založen na detekci teplotních špiček ve fotografii. Tato metoda první získala teplotní informace ze snímku, a poté je uspořádala do jednorozměrného pole. Převod fotografie do jednorozměrného pole proběhl postupným přidáváním řádků fotografie na



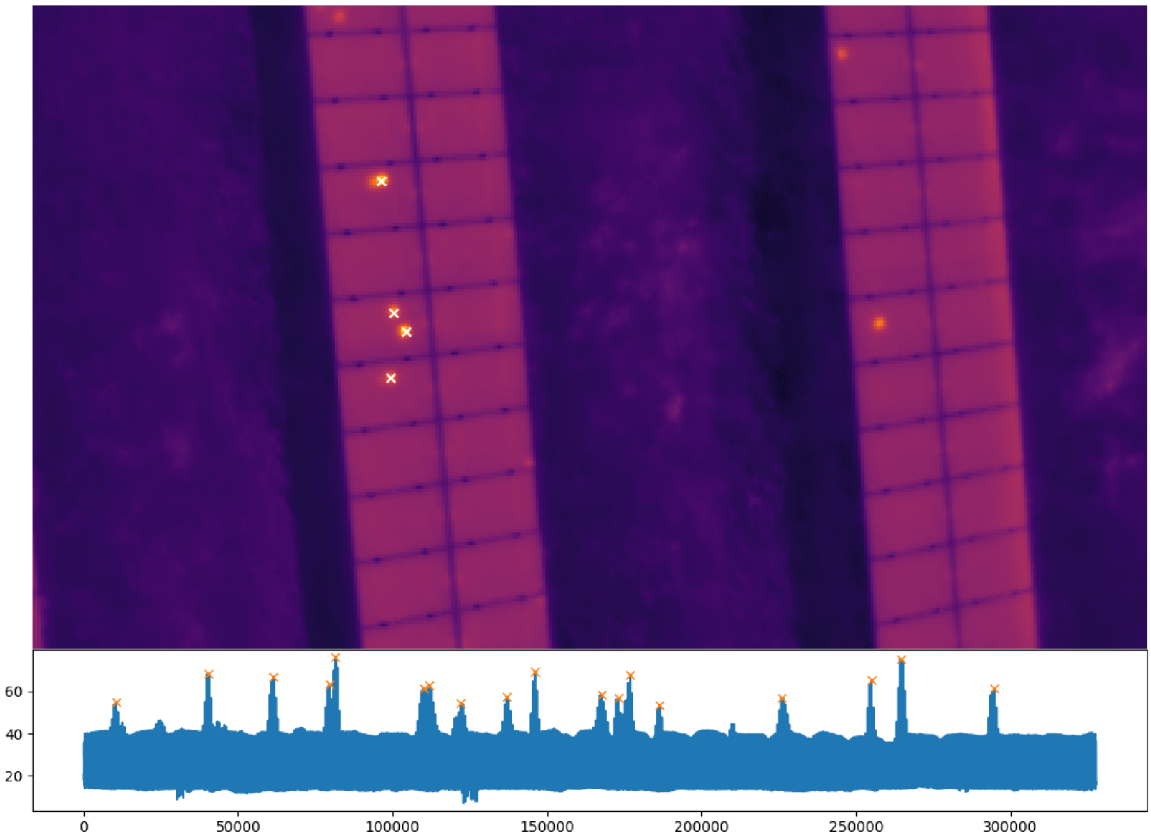
Obrázek 4.3: Původní fotografie (vlevo), výsledek extrakce panelů (vpravo).

konec daného pole. Tímto způsobem vznikla struktura obsahující teplotní hodnoty všech pixelů ve fotografii. Ve vzniklém poli se následně hledaly teplotní špičky. Parametry algoritmu pro vyhledávání špiček ve vstupním signálu mohou upravovat požadovanou vzdálenost mezi špičkami, strmost signálu v okolí špičky a různé další charakteristiky požadovaných špiček. Tento přístup se ale potýkal se stejnými problémy jako metoda prahování. Navíc způsobem tvoření jednorozměrného pole ze snímku vznikl nový problém, jedna chyba mohla být detekována hned několikrát. Důvodem je, že vzdálenost mezi dvěma pixely, které jsou ve fotografii přímo pod sebou, je rovna šířce fotografie. Je tomu tak, protože mezi těmito hodnotami jsou všechny hodnoty za horním pixelem, a všechny hodnoty před spodním pixelem. To znamená, že jedna chyba vytvoří hned několik teplotních špiček, které mají velmi podobné charakteristiky, ale mají mezi sebou značnou vzdálenost. Tento problém nelze vyřešit laděním parametrů pro algoritmus detekce špiček, protože by došlo ke kompromisu při detekci skutečných závad. Tento problém by se dal řešit například sloučením blízkých chyb, ale nelze zaručit, že by tohle řešení nevyloučilo i některé skutečné chyby, které by byly pouze blízko u sebe. Ukázka výstupu tohoto přístupu je na obrázku 4.4.

S ohledem na neuspokojivé výsledky triviálních přístupů a zohledněním trendů výzkumu v této oblasti [11] jsem se rozhodl použít pro řešení problému strojové učení a neuronovou síť.

4.2 Neuronová síť

Jako první jsem pro hledání závad využil přístup detekce objektů. V úkolu detekce objektů model neuronové sítě vrací obalový obdélník (z anglického „bounding box“), který ohraničuje nalezené objekty. Zvolil jsem architekturu neuronové sítě **Faster R-CNN** [17]. Pro otestování validity tohoto přístupu jsem vytvořil anotovanou datovou sadu pouze pro jeden typ závady - hot spot. Pro vyhodnocení výkonnosti modelu jsem použil metriku střední průměrná přesnost (z anglického „mean average precision“, dále mAP), blíže popsanou v sekci 6.1. Společně s trénovací datovou sadou, která obsahovala 300 anotovaných fotografií jsem vytvořil i testovací datovou sadu. Při trénování jsem vyzkoušel několik optimalizačních algoritmů a hyperparametrů. Pro vynahrazení nízkého počtu snímků v datové sadě jsem využil augmentace vstupních fotografií, a předem natrénovaný model knihovny



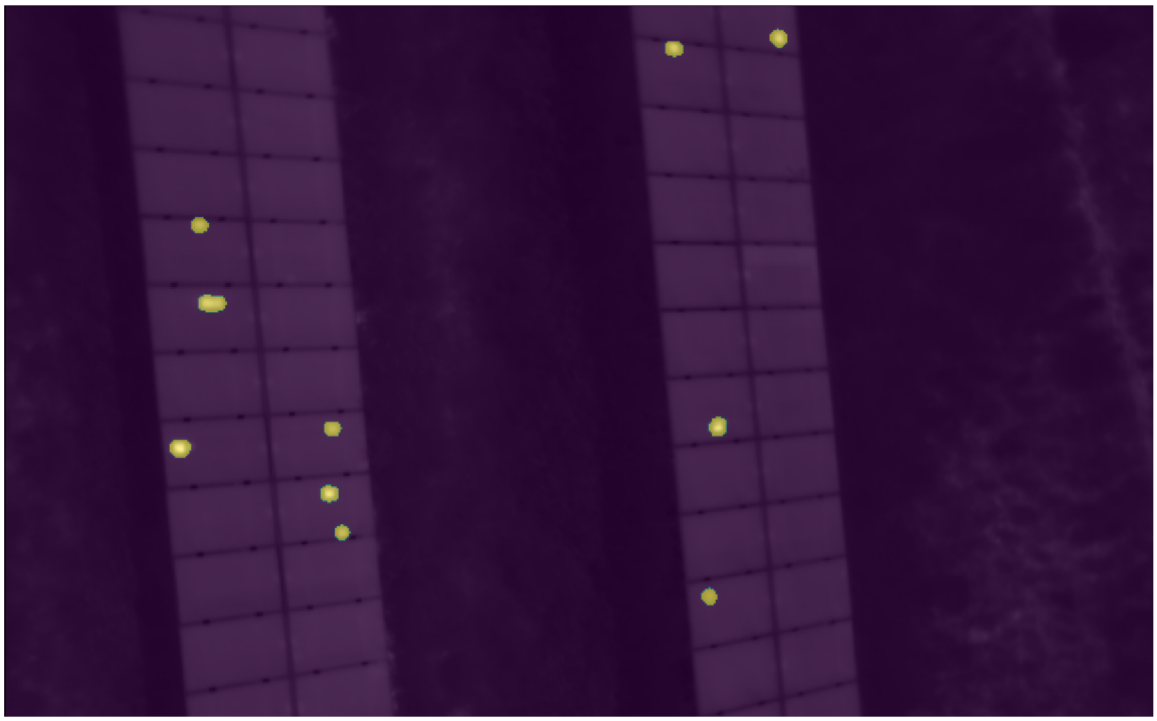
Obrázek 4.4: Detekce teplotních špiček ve snímku. V horní části je samotná fotografie, ve které jsou nalezené vady označeny bílými křížky. Ve spodní části je graf obsahující teplotní hodnoty všech pixelů. Osa x značí pořadí pixelu (číslo řádku × počet prvků v řádku + pořadí v řádku) a osa y značí teplotu daného pixelu.

„PyTorch“[1]. Model s nejlepší přeností ovšem dosáhl pouze 35 % mAP (viz sekce 6.2). Po vyhodnocení těchto výsledků a konzultace s odborníky jsem usoudil, že použitý model není vhodný na řešení problému, kde se hledají velmi malé anomálie ve fotografii.

Nově jsem zvolil přístup sémantické segmentace. Úkol sémantické segmentace přiřadí každému pixelu fotografie třídu, do které patří. Pro řešení tohoto úkolu jsem zvolil architekturu **U-Net** [18], dříve použitou k detekci závad na úrovni jednotlivých panelů [16]. U-Net je plně konvoluční neuronová síť, což znamená, že se skládá pouze z konvolučních vrstev. Původně byla navržena pro segmentaci zdravotnických snímků, například snímky magnetické rezonance. Forma anotací pro sémantickou segmentaci jsou masky pro jednotlivé snímky. Masky jsou obrázky stupně šedi o stejném tvaru jako anotovaná fotografie. Každá intenzita v masce představuje jednu třídu (např. všechny pixely o intenzitě 1 reprezentují vadu typu hot spot). Výstupem modelu pro segmentaci je také maska, přiřazující každému pixelu v původní fotografii nějakou třídu. Architekturu U-Net jsem využil ve finální implementaci.

Součástí práce je i detekce panelů, pro kterou jsem po vzoru [22] použil architekturu **Mask R-CNN**. Podstata detekce panelů je vysvětlena v sekci 4.3.

[1] https://pytorch.org/



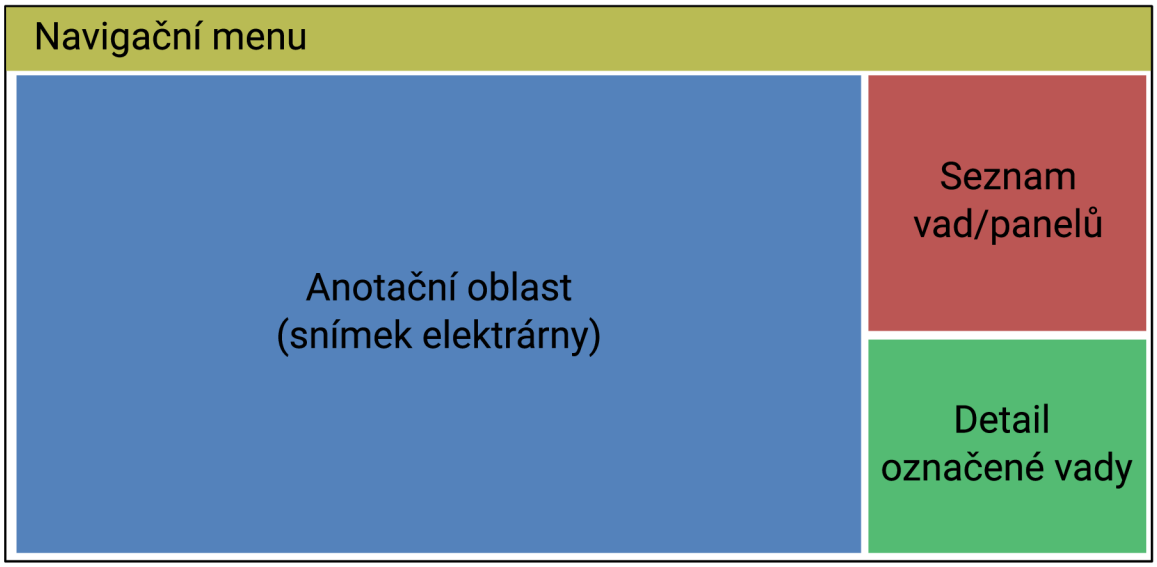
Obrázek 4.5: Výstup modelu *U-Net*. Fialová barva značí pozadí a žlutě jsou označeny nalezené závady typu hot spot.

4.3 Aplikace

Pro prezentaci výstupů vytvořeného detektoru chyb jsem vytvořil uživatelskou webovou aplikaci. V této aplikaci lze vytvářet jednotlivé anotace fotovoltaických elektráren. Do anotace je nahrána fotomozaika, obsahující poskládanou fotografii celé elektrárny. Následně lze elektrárna prohlížet, a označovat v ní jednotlivé chyby, tvořit nové chyby, nebo mazat existující. Návrh části aplikace pro manipulaci se závadami lze vidět na obrázku 4.6. Hlavní funkcí aplikace je automatická detekce, která se pokusí detekovat všechny závady v elektrárně. Samotný model, který provádí detekci závad lze jednoduše vyměnit za jiný, za předpokladu že nový model vrací výsledky v požadovaném formátu, což je maska, ve které každý pixel náleží určité třídě, určené podle intenzity pixelu (viz obrázek 4.5).

Další důležitou částí urychlení procesu inspekce je poskytnutí způsobu, jak určit pozici závady v kontextu celé elektrárny. Jak již bylo zmíněno v sekci 2.2 je za tímto účelem určeno, na kterém panelu se vada nachází. Pro identifikaci panelu je použito několik vlastností, které lze v aplikaci přiřadit skupině vybraných panelů hromadně. Vzhledem k tomu, že každá elektrárna může využívat jiný systém pro označení jednotlivých panelů, řad atd. nelze vytvořit systém pro automatické označení panelů. Z tohoto důvodu lze také přepnout do módu anotace panelů, ve kterém lze vybrat skupinu panelů ve fotomozaice a anotovat je. Vlastnosti panelu, které lze v aplikaci anotovat jsou část elektrárny, číslo řady a číslo lavice. Při tvorbě výsledné tabulky jsou využity tyto anotace panelů, a k dané chybě jsou informace o panelu doplněny, což jednoznačně určuje umístění vady. Před tím, než je možné panely anotovat, je nutné je ve fotomozaice nalézt. Využil jsem neuronovou síť pro detekci panelů, aby nemusel uživatel označovat všechny panely ručně.

Očekávaný postup při využití aplikace je:



Obrázek 4.6: Návrh hlavní anotační stránky

1. Vytvoření inspekce (pojmenování, nahrání fotomozaiky).
2. Vytvoření chybějících panelů, které detektor panelů neidentifikoval.
3. Využití detektoru závad.
4. Kontrola nalezených vad (mazání špatných detekcí a přidání chybějících).
5. Anotace panelů
6. Export výsledku inspekce

Aplikace také umožňuje přístup k jednotlivým detektorům i bez využití anotace na fotomozaice. K využití stačí nahrát fotografie a zvolit, který detektor využít (vad, nebo panelů). Následně budou nahrané snímky zobrazeny s nalezenými vadami a panely (podle toho, který detektor byl zvolen).

Kapitola 5

Implementace

V této kapitole popisuji, jakým způsobem jsem implementoval můj návrh z kapitoly 4. Řeší se zde použité technologie, popisuji architekturu mého kódu a také odkazuji na již existující řešení, které jsem během implementace využil.

5.1 Model pro detekci vad

Všechny modely neuronových sítí jsem implementoval v jazyce *Python* a frameworku *Pytorch*. Výsledky trénování i vyhodnocení všech neuronových sítí lze nalézt v kapitole 6.

Jako první jsem implementoval variantu modelu pro detekci objektů **Faster R-CNN**. Využil jsem implementaci modelu Faster R-CNN s „ResNet-50-FPN backbone“ dostupnou v knihovně Pytorch[1]. Pro trénování jsem využil tzv. „Transfer learning“, kdy trénování na vlastní datové sadě začalo s již natrénovaným modelem, a další učení pouze ladilo natrénovaný model pro mou datovou sadu (blíže popsáno v sekci 3.3). Datovou sadu pro detekci objektů (viz obrázek 5.1) jsem vytvořil ve formátu „COCO“[2]. Jednotlivé závady jsou reprezentovány ohraničujícími obdélníky, jejichž pozice, rozměry a náležící třída jsou popsány v náležícím souboru formátu JSON[3]. Tento model jsem trénoval na datové sadě o velikosti 300 fotografií, kterou jsem rozšířil augmentací vstupních fotografií. Specificky jsem použil augmentace: náhodná změna velikosti (až o 20%), vertikální i horizontální zrcadlení fotografie (s 50% šancí) a náhodná rotace.

Druhou architekturou neuronové sítě, kterou jsem pro tuto práci implementoval je **U-Net**. Použil jsem veřejně dostupnou implementaci[4] s mírnými úpravami pro mé účely. Tato implementace se drží původního návrhu architektury U-Net z [18], kterou lze vidět na obrázku 5.2. Pro trénování jsem využil prostředí „Google Colab“[5] s grafickou kartou NVIDIA Tesla V100. U tohoto modelu jsem experimentoval s řadou hyperparametrů a přístupů. Použitá datová sada obsahovala 773 anotovaných fotografií (viz obrázek 5.3) a využil jsem augmentace horizontální i vertikální zrcadlení (s 50% šancí) a náhodná rotace.

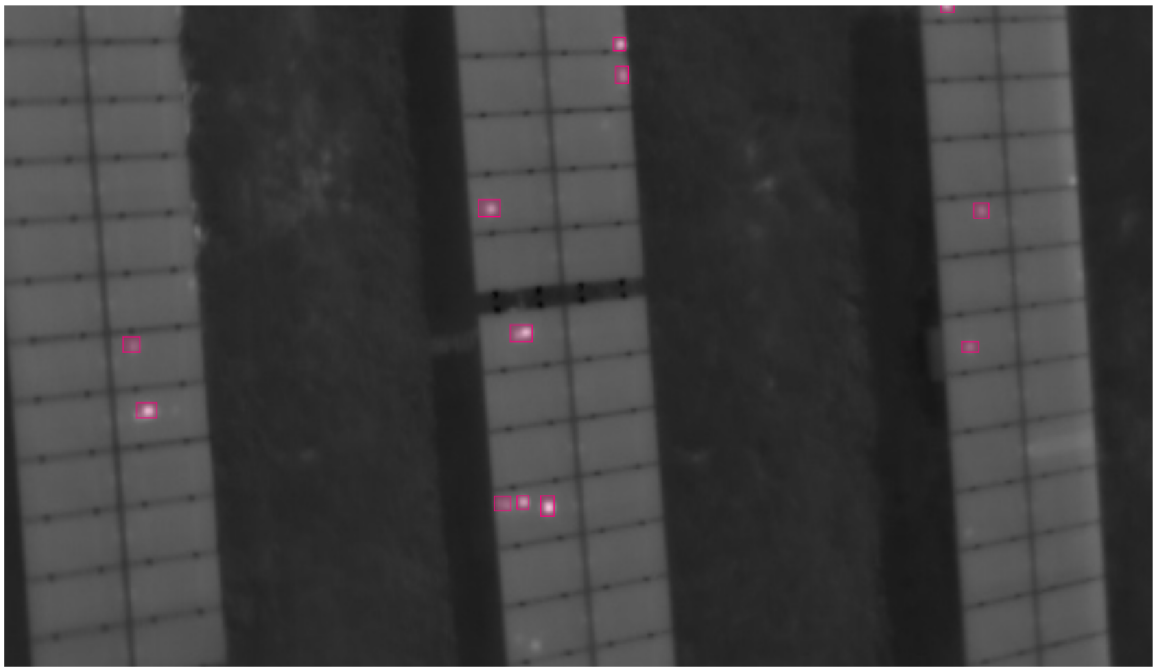
[1] https://pytorch.org/vision/main/models/generated/torchvision.models.detection.fasterrcnn_resnet50_fpn_v2.html

[2] z anglického „Common Objects in Context“

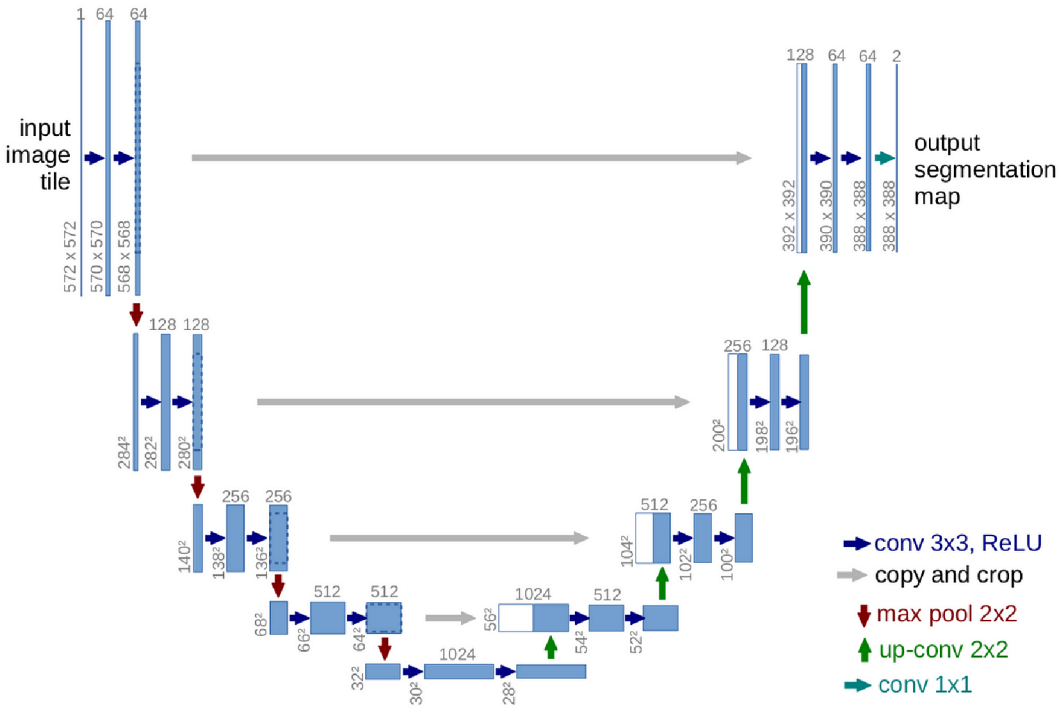
[3] z anglického „JavaScript Object Notation“

[4] https://github.com/milesial/Pytorch-UNet

[5] https://colab.research.google.com/



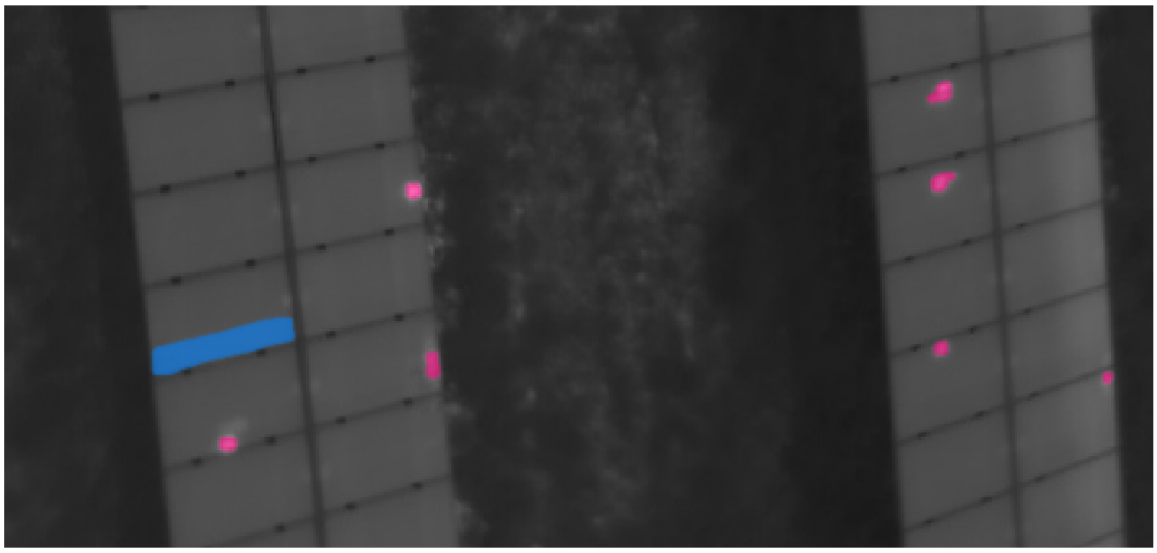
Obrázek 5.1: Anotovaný snímek použitý pro trénování Faster R-CNN. Růžové obdélníky jsou manuálně anotované vady typu hot spot.



Obrázek 5.2: Architektura modelu U-Net, převzato z [18].

Problémy s datovou sadou

Během anotování datové sady bylo odhaleno několik problémů s daty, kvůli kterým se přešlo pouze na úkol detekce bodů se zvýšenou teplotou. Typ závady nefunkční řada, která

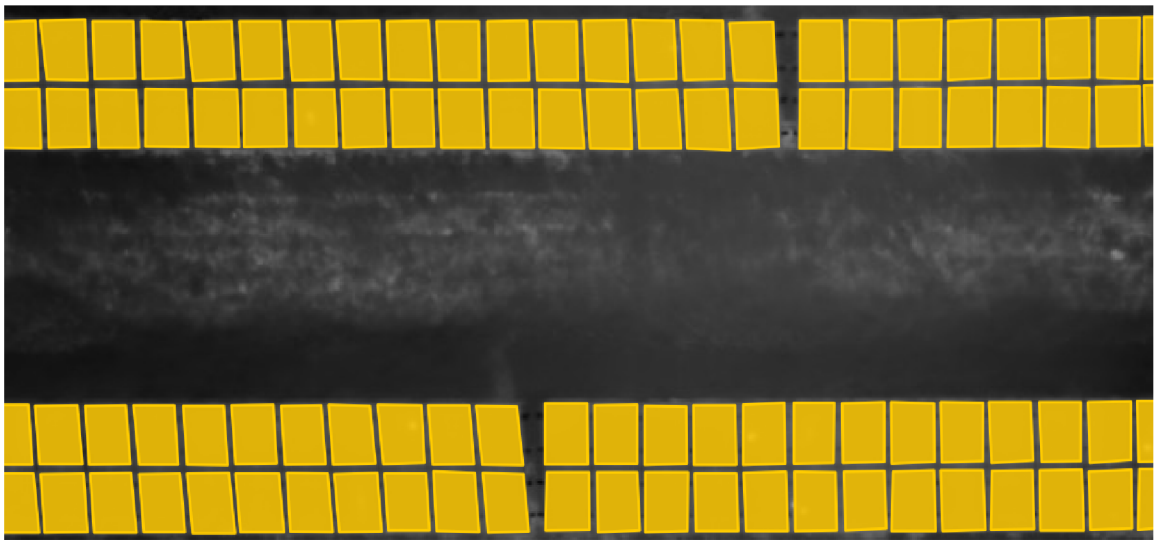


Obrázek 5.3: Anotovaný snímek pro trénování *U-Net* k sémantické segmentaci. Ručně vytvořené anotace značí vady typu hot spot (růžové) a nefunkční třetinu (modré).

se projevuje zvýšenou teplotou několika panelů v řadě se v poskytnutých nacházel pouze v pár ojedinělých případech. Naopak odrazy slunce (viz obrázek 4.1) se na panelech mohou projevit právě jako tento typ závady. Tyto odrazy ale nešlo využít k nahrazení této závady, protože často pokrývají pouze část panelu. Odrazy způsobují problém i s detekcí závady nefunkční třetiny panelu. Odrazy jsou totiž značně výraznější a v některých případech mohou mít tvar podobný této závadě. Finální model tedy skutečné závady tohoto typu téměř vždy ignoroval. Problém je i s detekcí zeleně ze snímku, jelikož se po konzultaci s profesionálem momentálně provádějícím inspekce FV elektráren došlo k závěru, že z termální fotografie nelze definitivně určit, jestli se jedná o hot spot, nebo právě zeleň. Obě vady se totiž projevují zvýšenou teplotou na části panelu. Pro definitivní určení je třeba porovnat tepelný i barevný snímek (je-li dostupný).

5.2 Model pro detekci panelů

Pro detekci panelů jsem použil model neuronové sítě s architekturou **Mask R-CNN** [5] implementovaný v knihovně Detectron2[6]. Model jsem metodou „Transfer learning“ (viz sekce 3.3) trénoval na datové sadě se 100 anotovanými fotografiemi (viz obrázek 5.4). Vstupní data jsem augmentoval horizontálním i vertikálním zrcadlením s 50 % pravděpodobností a rotací o 90° s pravděpodobností 50 %. *Mask R-CNN* provádí segmentaci instancí, tedy pro každou nalezenou instanci dané třídy vytvoří vlastní masku. Pro účely mé aplikace ovšem stačí pouze množina bodů, ohraničujících tvar panelu. Pro převod pixelových masek na množinu bodů jsem vytvořil skript, který ve fotografii detekuje tvary (masky panelů jsou na černém pozadí) a aproximuje jejich obrys množinou bodů.



Obrázek 5.4: Anotovaný snímek pro trénování *Mask R-CNN* k segmentaci instancí. Žluté obdélníky jsou manuálně vložené anotace značící panely ve fotografii.

5.3 Aplikace

Implementace aplikace se dá rozdělit na dvě hlavní části, a to „backend“ (práce s daty, logika aplikace) a „frontend“ (grafický výstup do uživatelského rozhraní). Tyto části spolu komunikují pomocí HTTP požadavků. Díky této architektuře je možné výpočetně náročné části programu přesunout na server. Pro účely této práce jsem spouštěl obě části lokálně.

HTTP Požadavky

Pro komunikaci mezi backend částí a frontend částí jsou používány zprávy protokolu HTTP. HTTP požadavky odesílá klient (frontend) a obsahují metodu (GET, POST, PUT, DELETE, atd.), cestu (v mém případě URL jednoho z koncových bodů), hlavičky a pro určité metody i tělo. Význam metod si udává aplikační rozhraní, populární je schéma ve kterém:

- GET slouží pro získávání informací.
- POST slouží ke vkládání nových informací.
- PUT slouží k aktualizaci existujících informací.
- DELETE slouží k odstranění existujících informací.

Pokud server (backend) obdrží požadavek ve správném formátu, provede se určitá akce, jejíž výsledkem je HTTP odpověď odeslána původnímu odesílateli. V odpovědi je místo metody status kód a zpráva, jinak má stejný formát jako požadavek[7]. Informace v těle HTTP zpráv jsou přenášeny ve formátu JSON.

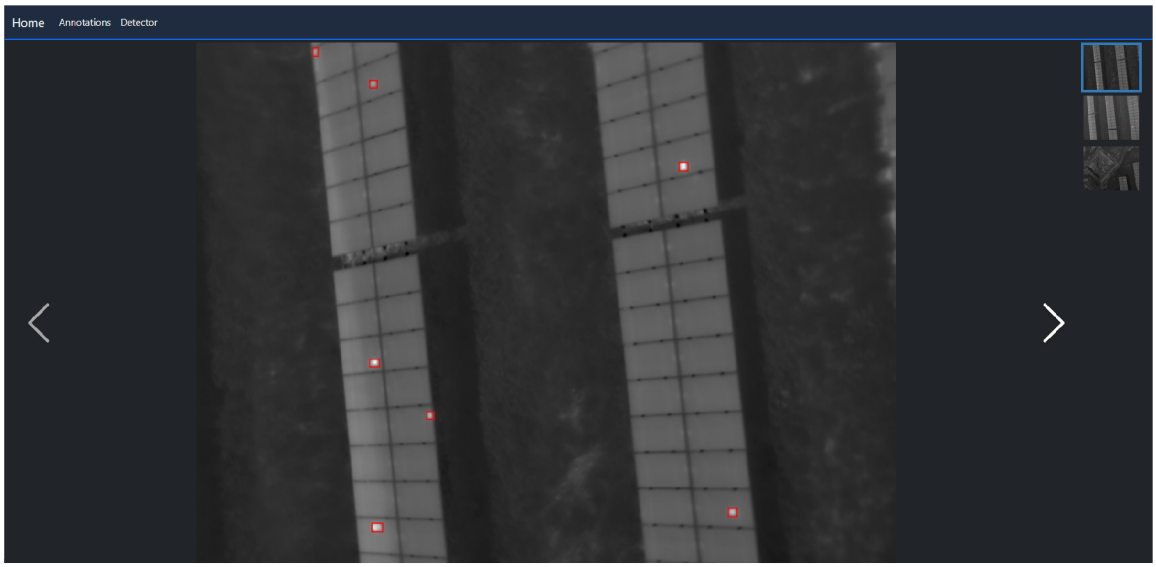
[6] *Mask R-CNN* s *ResNet-101+FPN backbone*
[7] https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview

Obrázek 5.5: Anotační stránka v aplikaci. Stránka zobrazuje fotomozaiku elektrárny s nalezenými vadami (nebo panely, dle vybraného režimu). Napravo je seznam vad, pod kterým jsou vypsané informace o uživatelem označené vadě. Označená vada je na fotomozaice zvýrazněna bílým kolečkem. Na levé straně fotomozaiky jsou tlačítka pro přepínámí režimu (zobrazení vad/panelů, výběr/tvorba).

Backend

Backend je implementován v jazyce Python. Tato volba je opodstatněná použitím knihovny „PyTorch“ pro využití natrénovaného detektoru chyb. Aplikační rozhraní (API) umožňující předávání výsledků pomocí protokolu HTTP je implementováno knihovnou *FastAPI*. Tato část aplikace odpovídá za kompletní práci s daty (ukládání, načítání) a implementaci logiky aplikace. Komunikaci s aplikačním rozhraním umožňují koncové body, které určité adrese (dále URL[8]) aplikačního rozhraní přiřazují příslušnou funkci, která se volá při přijetí požadavku na dané URL. Koncové body jsou rozděleny do vlastních souborů podle toho, s jakou částí aplikace souvisí (koncové body pracující s vadami jsou v jednom souboru, koncové body pracující s panely jsou v jiném souboru). Jednotlivé koncové body aplikačního rozhraní lze vidět na obrázku 5.8. Ve využité knihovně *FastAPI* jsou tyto koncové body zaregistrovány do směrovače („Router“) a jednotlivé směrovače jsou registrovány v hlavním API skriptu, který zajišťuje inicializaci a provoz celého aplikačního rozhraní. Funkce v koncových bodech definují chování aplikačního rozhraní. Pro práci s daty (načítání, ukládání) využívají repozitáře a pro detekci využívají detektory. Jak jsou jednotlivé části aplikačního rozhraní propojeny lze vidět na obrázku 5.7. Repozitář je návrhový vzor, který manipuluje data a zajišťuje transakční zpracování. Podobně jako směrovače jsou repozitáře rozděleny podle toho, s jakými daty pracují (panely, vady, anotace). Využití návrhového vzoru repozitář umožňuje snadnou změnu přístupu k ukládání dat, jelikož stačí vyměnit využité repozitáře a zbytek kódu může zůstat nezměněný. Detektory jsou implementovány jako funkce v Python skriptech. Hlavní funkcionalita, kterou umožňuje jak detektor vad, tak detektor panelů, je detekce vad (nebo panelů) v předané fotografii za využítím natrénovaného

[8]Uniform Resource Locator



Obrázek 5.6: Stránka pro detekci závad v jednotlivých snímcích.

modelu neuronové sítě. Druhá funkcionalita je detekce vad, nebo panelů, ve velké fotografii, za použití algoritmu posuvného okna, který je popsán níže.

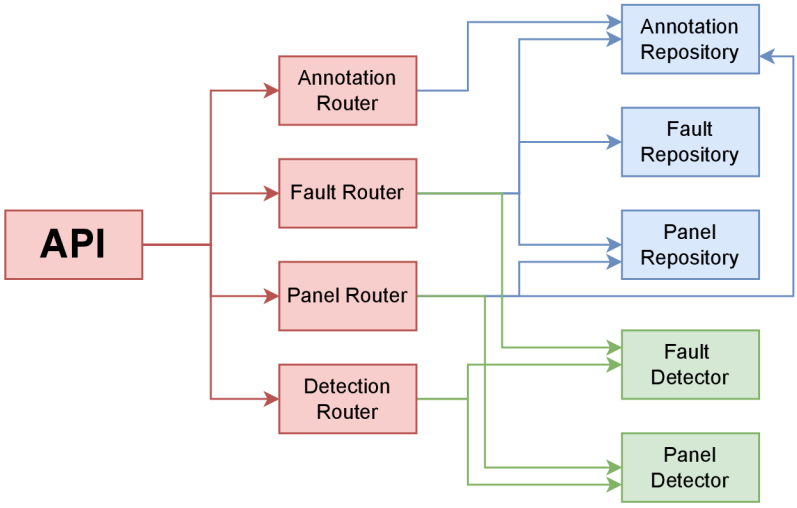
Frontend

Frontend je implementován pomocí knihovny *React* v jazyce TypeScript. Uživatelské rozhraní lze zobrazit ve webovém prohlížeči, díky čemuž je toto řešení multiplatformní. Všechna data (anotace, závady, panely) do této části aplikace přichází v odpovědích na HTTP požadavky zaslané frontend částí na backend část. Aplikace je rozdělena do několika stránek, které se skládají z React komponent z knihoven, nebo vlastně vytvořených. Mezi stránkami lze navigovat pomocí react směrovače. Ve hlavní stránce aplikace je zobrazena jedna anotace, což v kontextu této aplikace označuje celek obsahující fotografii elektrárny, označené panely a popřípadě anotované závady. Frontend obsahuje následující stránky:

- Seznam anotací.
- Tvorba anotace.
- Úprava anotace.
- Detekce.

Stránkou úprava anotace je myšlena hlavní anotační stránka (viz obrázek 5.5), kde se zobrazují a přidávají závady. Stránka „Detekce“ (viz obrázek 5.6) umožňuje využít detektory panelů a vad na nahrané obrázky, bez tvorby kompletní inspekce.

Závady lze na této stránce vytvářet přepnutím do režimu tvorby chyb. V tomto režimu lze táhnutím myši vytvořit box obsahující závadu. Typ závady je třeba určit před vytvořením chyby v bočním menu pod seznamem existujících chyb. Každá nově vytvořená chyba je hned při vytvoření uložena (odesláním požadavku na backend). Druhý dostupný režim slouží pro výběr vad. Po kliknutí na vadu se zobrazí detail s informacemi o dané vadě. Vady lze také vybrat hromadě tahem myši, v tomto případě jsou zvoleny všechny vady v nakresleném obdélníku a místo detailu vady je zobrazen seznam všech vad. Vedle mapy



Obrázek 5.7: Backend architektura. Červené části znázorňují koncové body. Modré části jsou zodpovědné za práci s daty a zelené části implementují logiku aplikace.

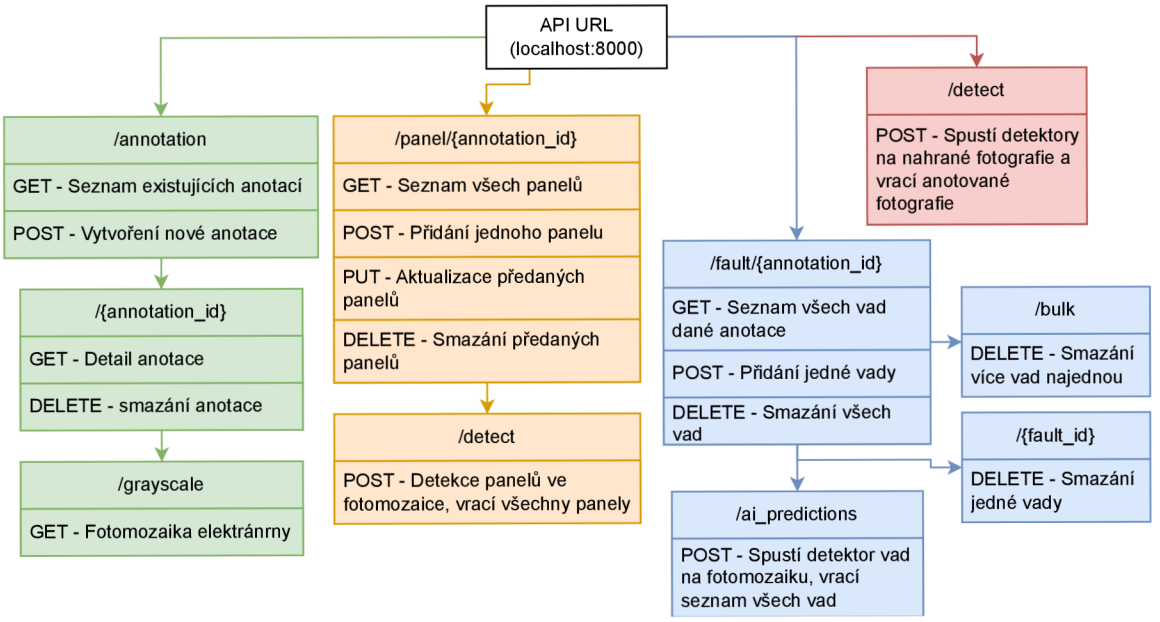
elektrárny je také viditelný seznam všech vad v dané anotaci a kliknutím na vadu v tomto seznamu se zvolí a obrázek se na ni vycentruje a přiblíží. Při přejetí kurzorem přes vadu v tomto seznamu se zvýrazní na mapě vykreslením kruhu okolo dané vady.

Automatická detekce závad proběhne na fotomozaice celé elektrárny. Vzhledem k rozměrům takové fotografie ji nelze dát jako vstup neuronové síti. Z mozaiky jsou proto vystřiženy fotografie o menších rozměrech přístupem posuvného okna. Aby detekce vad ve výstřižcích byla co nejpřesnější vybral jsem rozměr 640×512 pixelů, což je rozlišení snímků použitých pro trénování detektoru. Vady na okrajích obrazovky nemusí být správně detekovány, a proto je velikost kroku posuvného okna 3/4 velikosti okna. Celý postup vypadá následovně:

1. Algoritmus posuvného okna vytvoří výstřižek, neboli okno (začíná vlevo nahoře).
2. Okno je použito jako vstup detektoru vad
3. Vady jsou uloženy pouze v případě, že se nepřekrývají s jinou vadou stejného typu[9] (Tento krok je nutno provést kvůli tomu, že velikost kroku je menší než velikost okna, což způsobí, že vady na okraji okna mohou být detekovány dvakrát.)
4. Pokud se okno může posunout o velikost kroku doprava, posune se a postup opakuje se od kroku 1
5. Pokud se už okno nemůže posunout vpravo (je na kraji fotografie), posune se úplně doleva a také se posune dolů o velikost kroku, poté se opakuje krok 1
6. Pokud se okno nemůže posunout ani vpravo, ani dolů, je detekce vad dokončena

Jak již bylo zmíněno v sekci 4.3, je důležité umožnit anotaci jednotlivých panelů v elektrárně. K tomu slouží přepínač, který určuje, jestli se zobrazují panely, nebo vady. Při zobrazení panelů je lze tvořit a mazat stejně jako vady. Při označení panelů lze vpravo dole anotovat jejich vlastnosti (část elektrárny, řada, lavice).

[9] Pro efektivní kontrolu překryvu je využita datová struktura *R-tree*



Obrázek 5.8: Koncové body aplikačního rozhraní. Zeleně koncové body pro anotace, zlutě koncové body pro panely, modře koncové body pro závady a červeně koncové body pro detekci.

Kapitola 6

Testování a vyhodnocení

Tato kapitola obsahuje výsledky testování mnou implementovaných neuronových sítí. Pro model *U-Net*, detekující závady, jsem vytvořil testovací sadu s třiceti anotovanými snímky. Testovací anotace mají stejnou podobu jako anotace pro trénování, které jsou popsané v sekci 5.1. Stejně tak jsem vytvořil i testovací datovou sadu s deseti snímky pro model *Mask R-CNN*, detekující panely ve fotografiích. Pro vyhodnocení výsledného výkonu detektorů jsem použil metriky, které jsou blíže popsané v sekci 6.1.

6.1 Metriky

Pro řešení problému této práce jsem využil více přístupů, u kterých jsem použil různé metriky pro měření výkonu výsledného modelu.

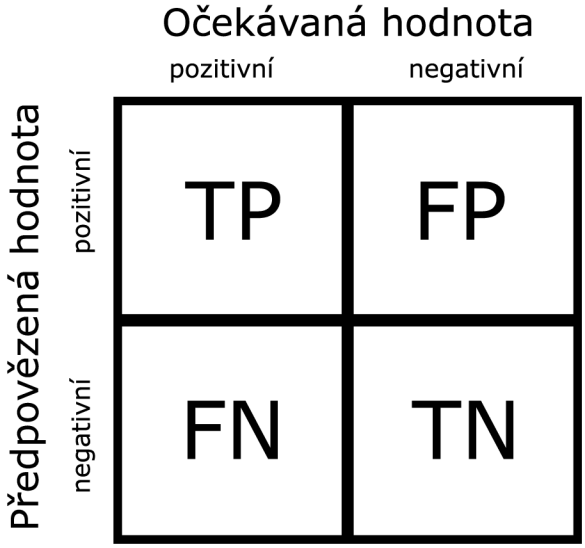
Mean Average Precision (mAP)

Při úkolu detekce objektů bylo hlavní metrikou „Mean Average Precision“ („mAP“). Pro vysvětlení tohoto pojmu je třeba první vysvětlit význam jistých pojmů, které tato metrika využívá. První důležitý pojem je „Confusion matrix“.

„Confusion matrix“ je matice, která obsahuje čtyři prvky. Umístění prvků v této matici udává co tyto prvky značí. Řádek matice udává hodnotu předpovězenou neuronovou sítí a sloupec udává očekávanou hodnotu (hodnoty jsou dvě, pozitivní a negativní). Pokud je očekávaná i předpovězená hodnota pozitivní, jedná se o „True Positive“ (dále TP) a pokud jsou obě hodnoty negativní, jde o „True Negative“ (dále TN). V obou těchto případech předpověděla neuronová síť hodnotu správně. Mohou ale nastat dva případy, ve kterých předpověď správně nebude. „False Positive“ (dále FP) kdy model předpověděl pozitivní hodnotu, ale byla očekávána negativní a opačně „False Negative“ (dále FN), kdy předpovězená hodnota byla negativní, ale očekávaná pozitivní. Matici lze vidět na obrázku 6.1.

Je nutné stanovit, jak se určuje jestli je předpovězená hodnota pozitivní nebo negativní. V kontextu detekce objektů je nutno pamatovat že pracujeme s ohraničujícími obdélníky. Při testování je nutné poskytnout obdélníky, které odpovídají pravdě (anglicky „ground truth“) a reprezentují nejlepší možnou odpověď. Pro každý tento obdélník je poté zkontrolováno, jestli některý předpovězený obdélník překrývá očekávaný obdélník. Je třeba stanovit hranici překryvu, která určuje, jestli se obdélníky překrývají dostatečně. Pro určení poměru překryvu dvou boxů slouží „Intersection over Union“ („IoU“).

$$\text{IoU} = \frac{\text{Obsah plochy překryvu}}{\text{Obsah plochy sjednocení}}$$



Obrázek 6.1: Confusion Matrix. Řada určuje předpovězenou hodnotu (pozitivní, nebo negativní) a sloupec určuje očekávanou hodnotu. Umístění v matici určuje charakter předpovědi. Na diagonále jsou správné předpovědi (TP – „True Positive“, TN – „True Negative“), které odpovídají skutečnosti. Ostatní prvky matice (FN – „False Negative“, FP – „False Positive“) značí nesprávnou předpověď.

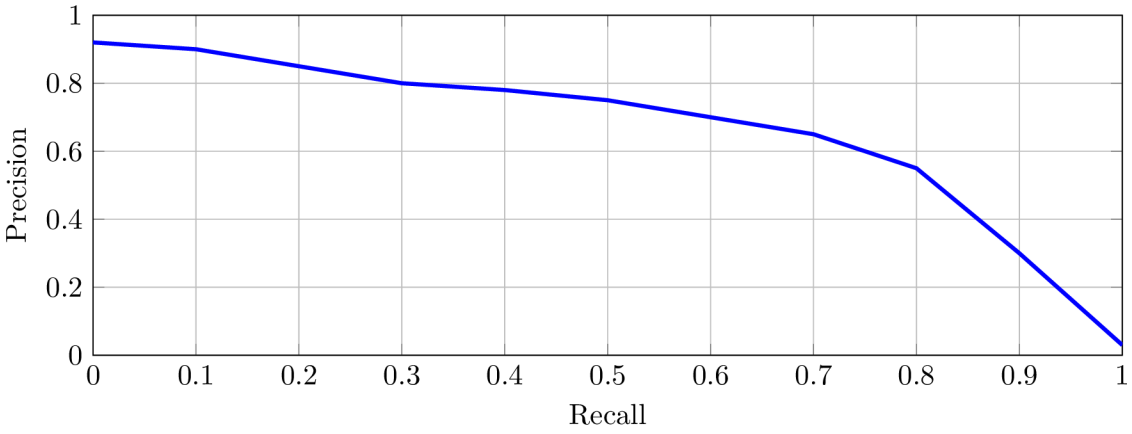
Z této rovnice plyne, že hodnoty IoU se mohou pohybovat v rozmezí $\langle 0, 1 \rangle$. Pokud se tedy předpovězený box překrývá s dostatečnou IoU se skutečným boxem, jedná se o TP. Pokud skutečný box není (dostatečně) překrýván předpovězeným boxem, jedná se o FN a pokud existuje předpovězený box, pro který neexistuje žádný skutečný box, jedná se o FP. TN není pro výpočet mAP relevantní (Populární přístup při výpočtu mAP je zprůměrovat výsledky mAP pro různé IoU hodnoty).

Hodnoty z „confusion matrix“ jsou využity pro výpočet metrik „Precision“ a „Recall“:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Z těchto rovnic lze odvodit význam vypočítaných metrik. „Precision“ určuje přesnost pozitivních předpovědí modelu, neboli jak důvěryhodné jsou pozitivní předpovědi daného modelu. Vysoká „Precision“ hodnota naznačuje, že model nepředpovídá mnoho falešných pozitiv. „Recall“ udává schopnost modelu najít všechny relevantní případy v datech. „Recall“ říká, kolik skutečně pozitivních případu model zachytil. To znamená, že vysoké hodnoty naznačují, že model je schopen najít většinu relevantních objektů.

V ideálním případě by obě hodnoty měli nejvyšší hodnotu (1), takového výsledku ovšem v téměř žádném reálném případu využití nelze dosáhnout. Je tedy třeba nalézt kompromis mezi těmito metrikami. V kontextu detekce objektů je každé předpovědi boxu přiřazena jistota (pravděpodobnost). Při zpracování výstupu je třeba určit hranici jistoty, nad kterou se předpověď počítá jako platná. Pro různé hodnoty této hranice budou získány různé hodnoty „Precision“ a „Recall“. Pokud se zakreslí vztah těchto hodnot do grafu, vznikne „Precision-Recall curve“ (viz obrázek 6.2).



Obrázek 6.2: Precison-Recall křivka.

Vypočtením obsahu pod touto křivkou se získá metrika „Average Precision“ (AP). Pokud model předpovídá více tříd, výpočtem průměru z AP každé třídy lze dostat „Mean Average Precision“ (mAP).[1]

Sørensen–Dice koeficient

Hlavní metrikou při vyhodnocení sémantické segmentace snímků je „Sørensen–Dice coefficient“ (dále Dice koeficient, Dice skóre). Dice koeficient je metrika podobnosti, kterou lze využít i na fotografie, což je užitečné při porovnávání výstupu neuronové sítě provádějící sémantickou segmentaci (pixelová maska s různými intenzitami dle třídy, viz obrázek 4.5) s očekávanou (člověkem vytvořenou) maskou. Lze vypočítat rovnicí:

$$\text{Dice koeficient} = \frac{2 * |A \cap B|}{|A| + |B|}$$

kde A, B jsou množiny, $|A|$ je počet prvků v množině A a $|A \cap B|$ je počet prvků v průniku množin A a B. Dice koeficient může nabývat hodnot $\langle 0, 1 \rangle$. Obecně tato metrika slouží k určení podobnosti libovolných množin, můžeme ji tedy použít pro porovnání obrázků, pokud bereme jako prvky množiny hodnoty jednotlivých pixelů v obrázku. Z toho plyne, že cílem při sémantické segmentaci je docílit co nejvyššího Dice koeficientu pro předpovědi modelu. Tento koeficient lze použít i při výpočtu účelové funkce, díky jeho dobrým vlastnostem při nevybalancovaných datech.[2]

6.2 Modely pro detekci vad

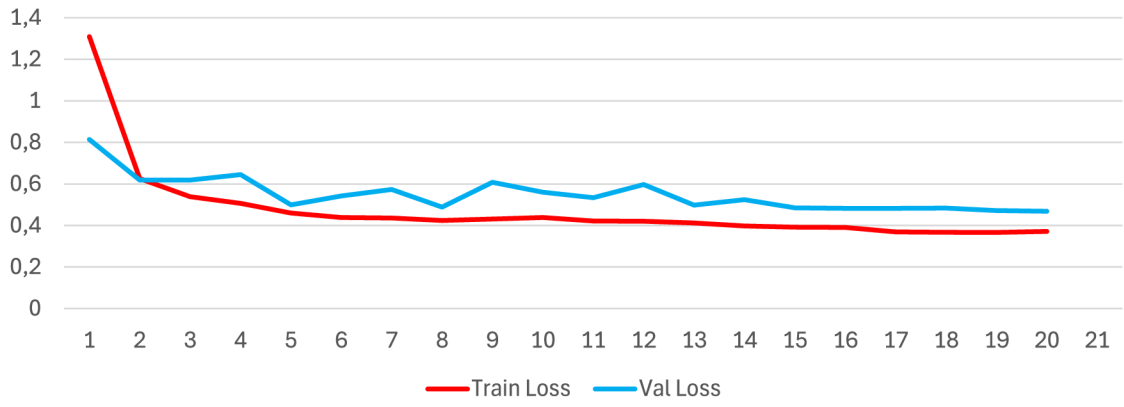
Faster R-CNN

Jak bylo již zmíněno v sekci 5.1, první jsem implementoval model *Faster R-CNN*.

[1] https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide

[2] https://medium.com/@lathashreeh/dice-coefficient-what-is-it-ff090ec97bda

Experiment 1
Model jsem trénoval 20 epoch[3], použil jsem optimalizační algoritmus *Adam*[4], lr[5] $= 1 * 10^{-3}$, kroková změna lr v 15. epoše s deltou $0,1$. Obrázek 6.3 obsahuje graf z trénování v tomto experimentu.



Obrázek 6.3: Graf z trénování *Faster R-CNN*.

Experiment 2
Model by trénován 20 epoch, lr $= 1 * 10^{-4}$, kroková změna lr v 10. epoše s deltou $0,1$.

lr	**mAP**	mAP (IoU 0.5)	mAP (IoU 0.75)
0,001	**0,3506**	0,9064	0,1681
0,0001	**0,3416**	0,8916	0,1201

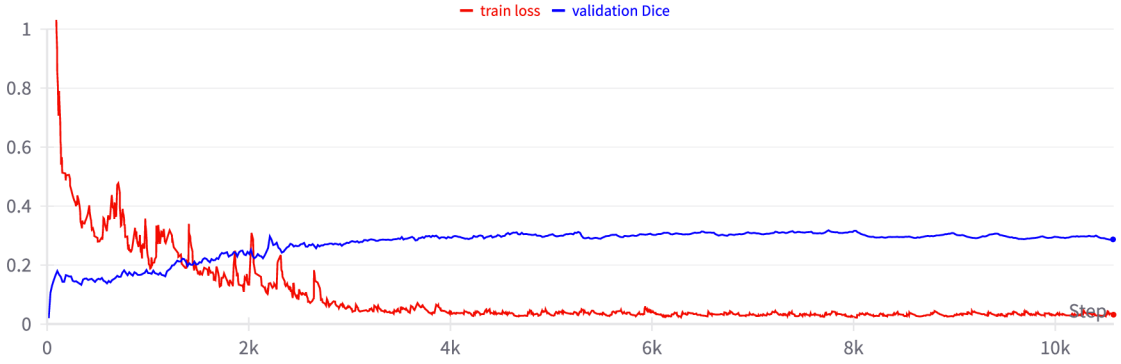
Tabulka 6.1: *Faster R-CNN* metriky

U-Net

Všechny modely v této sekci využívaly datovou sadu se 770 anotovanými fotografiemi pro sémentaickou segmentaci. Augmentace vstupních dat se skládaly z vertikálního a horizontálního zrcadlení (obě s pravděpodobností 50 %) a rotace o $\langle -90°, 90° \rangle$. Třídy (druhy závad) obsažené v této datové sadě závisí na jednotlivém experimentu a jsou u něj uvedeny. Účelová („loss“) funkce byla původně součet *Cross Entropy* a *Dice Loss*[6] pro první dva experimenty. Od třetího experimentu jsem využil pouze *Dice Loss* funkci. Výsledky modelů natrénovaných v jednotlivých experimentech lze vidět v tabulce 6.2

Experiment 1
Model jsem trénoval 120 epoch, lr $= 1 * 10^{-3}$, kroková změna lr každých 30 epoch s deltou $0,1$, optimalizační algoritmus *Adam*. Datová sada obsahovala třídy hot spot a nefunkční třetina (s váhami). Validační **Dice** skóre během trénování dosáhlo 0,2903. Graf z trénování lze vidět na obrázku 6.4.

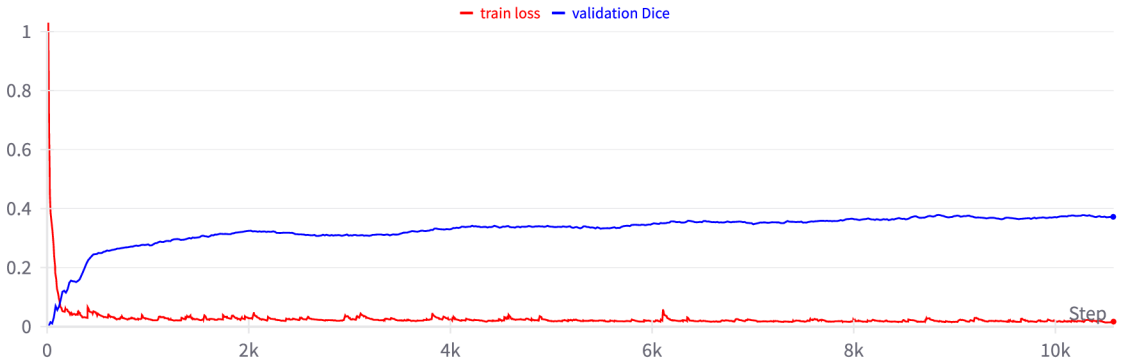
[3] Jedna epocha značí jeden průchod celou trénovací datovou sadou.
[4] https://pytorch.org/docs/stable/generated/torch.optim.Adam.html
[5] lr („learning rate“) značí velikost kroku optimalizačního algoritmu
[6] 1 – Dice koeficient



Obrázek 6.4: *U-Net* Experiment 1.

Experiment 2

Od tohoto experimentu jsem z trénovací datové sady odstranil třídu pro nefunkční třetinu panelu. Model jsem trénoval 120 epoch, lr $= 1 * 10^{-5}$, změna lr po pěti epochách se stagnujícím validačním Dice koeficientem[7] s deltou $0,1$, optimalizační algoritmus *RMSprop*[8]. Datová sada obsahovala třídy hot spot a nefunkční třetina (s váhami). Validační **Dice** skóre během trénování dosáhlo 0,3802. Graf z trénování lze vidět na obrázku 6.5.



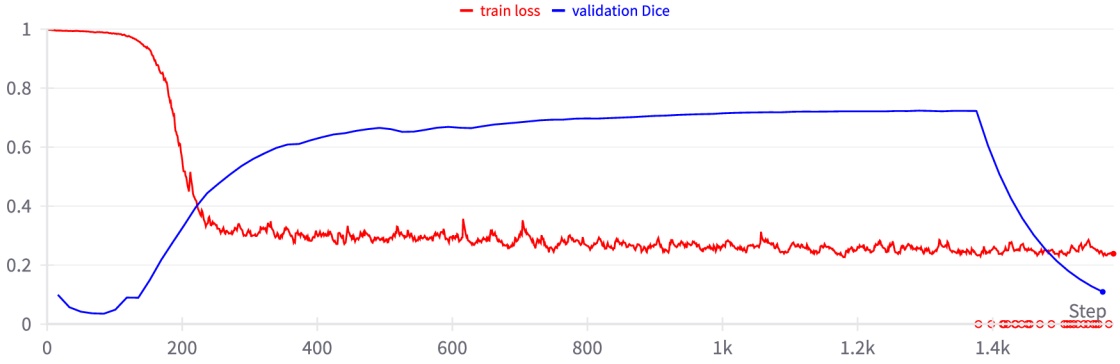
Obrázek 6.5: *U-Net* Experiment 2.

Experiment 3

V tomto experimentu jsem využil metodu „transfer learning“. Model jsem trénoval 18 epoch, lr $= 5 * 10^{-6}$, změna lr po stagnaci 10 epoch s deltou $0,1$, optimalizační algoritmus *RMSprop*. Datová sada obsahovala pouze závady typu hot spot. Tento experiment jsem zastavil předčasně, protože hodnoty účelové funkce začaly nabývat neplatných hodnot. Pro zamezení tomuto problému jsem v následujícím experimentu zvolil jiný optimalizační algoritmus a nevyužil jsem metody „transfer learning“. Validační **Dice** skóre během trénování dosáhlo 0,7314. Graf z trénování lze vidět na obrázku 6.6.

[7] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

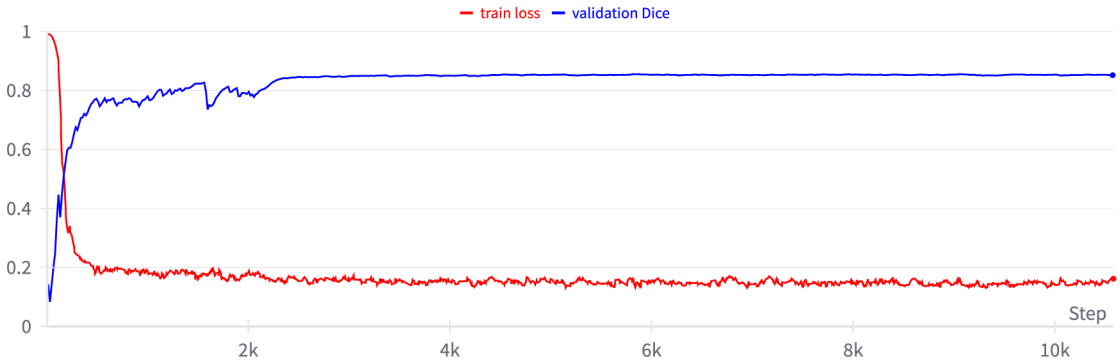
[8] https://pytorch.org/docs/stable/generated/torch.optim.RMSprop.html



Obrázek 6.6: *U-Net* Experiment 3.

Experiment 4

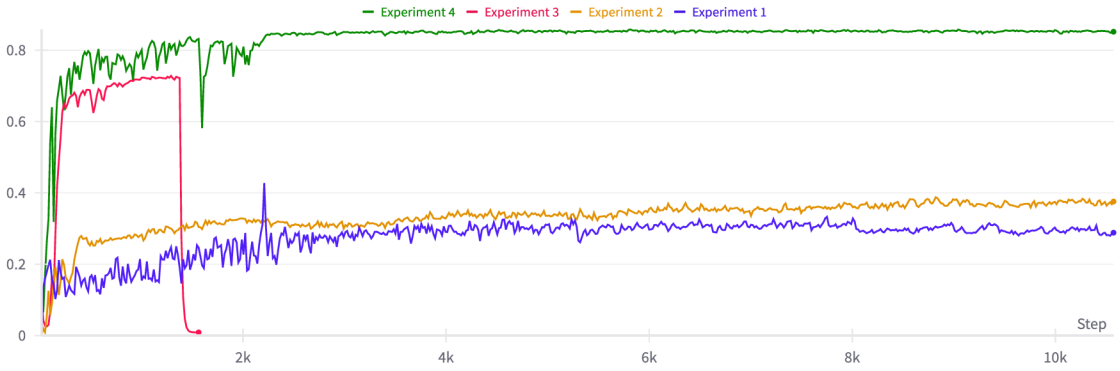
Model jsem trénoval 120 epoch, lr $= 1 * 10^{-3}$, kroková změna lr každých 25 epoch s deltou $0,1$, optimalizační algoritmus *Adam*. Datová sada obsahovala pouze závady typu hot spot. Tento experiment dosáhl nejlepších výsledků. Validační **Dice** skóre během trénování dosáhlo 0,8532. Graf z trénování lze vidět na obrázku 6.7.



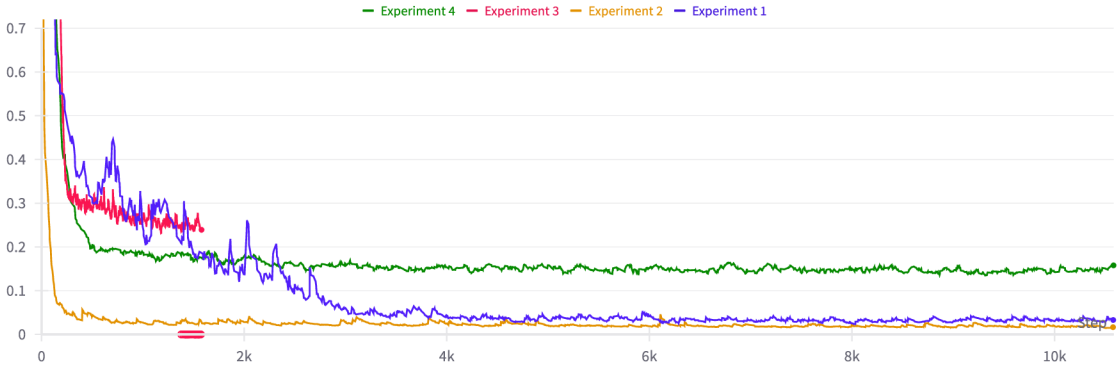
Obrázek 6.7: *U-Net* Experiment 4.

Porovnání výsledků

Graf v obrázku 6.8 obsahuje porovnání validačního dice skóre během trénování jednotlivých modelů a graf v obrázku 6.9 obsahuje hodnoty účelové funkce jednotlivých experimentů. Pro získání výsledků v tabulce 6.2 jsem porovnal výstupy natrénovaných modelů nad vytvořenými anotacemi v testovací datové sadě s třiceti anotovanými fotografiemi. Dva modely detekují dvě třídy, zatímco další dva pouze jednu, proto jsem pro modely, které detekují pouze body se zvýšenou teplotou, odstranil ostatní anotace z testovacích dat.



Obrázek 6.8: Porovnání *Dice* skóre modelů při trénování.



Obrázek 6.9: Porovnání hodnot *Loss funkce* modelů při trénování.

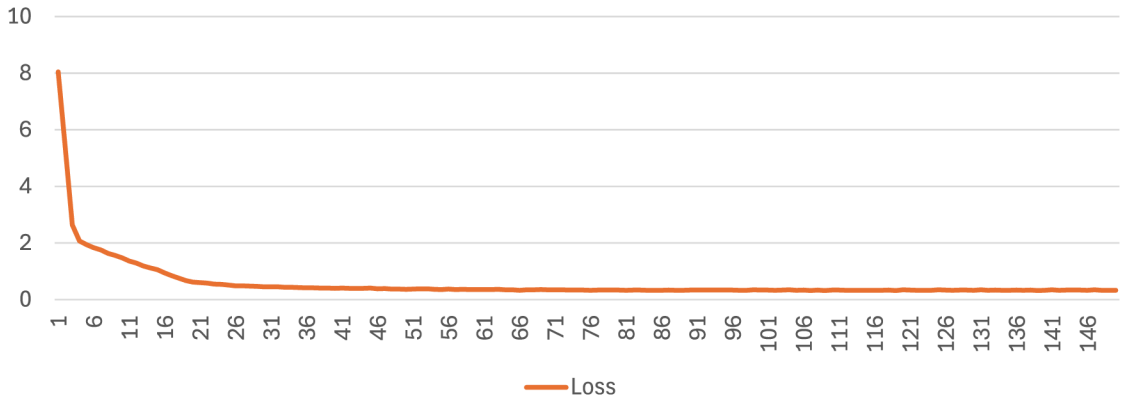
Experiment	**Dice**	Precision	Recall
1	**0,1563**	0,089	0,9403
2	**0,2077**	0,1199	0,9734
3	**0,6669**	0,7155	0,6527
4	**0,7037**	0,7301	0,6994

Tabulka 6.2: Porovnání natrénovaných *U-Net* modelů

6.3 Model pro detekci panelů

V knihovně Detectron2 se místo epoch využívají pro určení doby trénování iterace[9]. Model byl trénován 3000 iterací s velikostí dávky 8 snímků. Datová sada obsahovala 100 anotovaných fotografií a byla augmentována náhodným horizontálním i vertikálním zrcadlením a rotací o 90 stupňů s 50 % šancí. Výsledná mAP pro obalové obdélníky byla **0,8335** a pro masky **0,8275**. Graf z trénování lze vidět na obrázku 6.10.

[9]Jedna iterace odpovídá průchodu jedné dávky, neboli zpracování n obrázků, kde n odpovídá „Batch size“.



Obrázek 6.10: Graf loss funkce z trénování Mask R-CNN pro detekci panelů.

Kapitola 7

Závěr

Implementovaná aplikace má za cíl urychlit proces inspekce fotovoltaických elektráren, který momentálně trvá několik dní. Hlavními způsoby, jak toho docílit, jsou systém automatické detekce vad a navržení intuitivního prostředí pro přiřazení závad k panelům.

Detektor vad využívající neuronovou síť momentálně zvládá detekovat nejčastější typ závady – hot spot. Validační Dice skóre během trénování bylo 0,85 a na testovací datové sadě 0,7. Výkon modelu by mohl být zlepšen rozšířením datové sady, kvalitnějšími daty pro trénování (eliminace světelných odrazů na panelech), a také využitím novější architektury U-Net, popřípadě jiného moderního segmentačního modelu. S rozšířenou a kvalitnější datovou sadou by také mělo být možné detekovat, až na zeleň, všechny druhy závad.

Proces lokalizace fotovoltaických panelů v kontextu celé elektrárny byl navrženým prostředím zjednodušen, díky možnosti označovat panely hromadně. Další zlepšení oproti momentálnímu stavu spočívá ve využití fotomozaiky, což umožňuje vidět pozici panelu v elektrárně přímo během anotace. Model detekující panely ve snímku dosáhl mAP 0.83, což v kombinaci s přístupem posuvného okna dokáže ve fotomozaice fotovoltaické elektrárny detekovat většinu panelů. Uživatel tedy musí smazat, nebo vytvořit poměrně málo (v řádu desítek) panelů z celé elektrárny.

Vytvořená fotomozaika elektrárny neobsahuje teplotní informace, což je problematické při anotaci vady typu hotspot, kde nastává problém s určením kategorie dané vady. Nejjednodušším řešením by bylo zahrnout do ní teplotní informaci, ovšem samotná tvorba fotomozaiky nespadá do rozsahu této práce. Při využití aplikace popsané v této práci je stále třeba do určité míry ručně doplnit vytvořenou tabulku, ale značná část tabulky je předem vyplněná. Součástí výsledné anotace byly také anotované fotografie. Momentálně je výstupem aplikace celá fotografie anotované fotomozaiky. Aplikace by mohla být rozšířena algoritmem, který by detekoval shluky vad a vytvořil sadu výsledných snímků, podobně jak je tomu při manuální inspekci.

Literatura

[1] Akram, M. W., Li, G., Jin, Y., Chen, X., Zhu, C. et al. Automatic detection of photovoltaic module defects in infrared images with isolated and develop-model transfer deep learning. *Solar Energy.* Elsevier. 2020, sv. 198, s. 175–186.

[2] Dunderdale, C., Brettenny, W., Clohessy, C. a Dyk, E. E. van. Photovoltaic defect classification through thermal infrared imaging using a machine learning approach. *Progress in Photovoltaics: Research and Applications.* Wiley Online Library. 2020, sv. 28, č. 3, s. 177–188.

[3] Gade, R. a Moeslund, T. B. Thermal cameras and applications: a survey. *Machine vision and applications.* Springer. 2014, sv. 25, s. 245–262.

[4] Golovko, V., Kroshchanka, A., Bezobrazov, S., Sachenko, A., Komar, M. et al. Development of solar panels detector. In: IEEE. *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T).* 2018, s. 761–764.

[5] He, K., Gkioxari, G., Dollár, P. a Girshick, R. Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision.* 2017, s. 2961–2969.

[6] Herraiz, Á. H., Marugán, A. P. a Márquez, F. P. G. Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure. *Renewable Energy.* Elsevier. 2020, sv. 153, s. 334–348.

[7] J.A. Tsanakas, P. B. a Gasteratos, A. Fault diagnosis of photovoltaic modules through image processing and Canny edge detection on field thermographic measurements. *International Journal of Sustainable Energy.* Taylor & Francis. 2015, sv. 34, č. 6, s. 351–372. DOI: 10.1080/14786451.2013.826223. Dostupné z: https://doi.org/10.1080/14786451.2013.826223.

[8] Kardasz, P., Doskocz, J., Hejduk, M., Wiejkut, P. a Zarzycki, H. Drones and possibilities of their using. *J. Civ. Environ. Eng.* 2016, sv. 6, č. 3, s. 1–7.

[9] Kim, D., Youn, J. a Kim, C. Automatic photovoltaic panel area extraction from UAV thermal infrared images. *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography.* Korean Society of Surveying, Geodesy, Photogrammetry and Cartography. 2016, sv. 34, č. 6, s. 559–568.

[10] Li, B., Delpha, C., Diallo, D. a Migan Dubois, A. Application of Artificial Neural Networks to photovoltaic fault detection and diagnosis: A review. *Renewable and Sustainable Energy Reviews.* 2021, sv. 138, s. 110512. DOI:

https://doi.org/10.1016/j.rser.2020.110512. ISSN 1364-0321. Dostupné z: https://www.sciencedirect.com/science/article/pii/S136403212030798X.

[11] Meribout, M., Kumar Tiwari, V., Pablo Peña Herrera, J. a Najeeb Mahfoudh Awadh Baobaid, A. Solar panel inspection techniques and prospects. *Measurement*. 2023, sv. 209, s. 112466. DOI: https://doi.org/10.1016/j.measurement.2023.112466. ISSN 0263-2241.

[12] Nie, J., Luo, T. a Li, H. Automatic hotspots detection based on UAV infrared images for large-scale PV plant. *Electronics Letters*. Wiley Online Library. 2020, sv. 56, č. 19, s. 993–995.

[13] Nooralishahi, P., Ibarra Castanedo, C., Deane, S., López, F., Pant, S. et al. Drone-based non-destructive inspection of industrial sites: A review and case studies. *Drones*. MDPI. 2021, sv. 5, č. 4, s. 106.

[14] Parhar, P., Sawasaki, R., Todeschini, A., Vahabi, H., Nusaputra, N. et al. HyperionSolarNet: solar panel detection from aerial images. *ArXiv preprint arXiv:2201.02107*. 2022.

[15] Pierdicca, R., Malinverni, E., Piccinini, F., Paolanti, M., Felicetti, A. et al. Deep convolutional neural network for automatic detection of damaged photovoltaic cells. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Copernicus GmbH. 2018, sv. 42, s. 893–900.

[16] Rahman, M. R. U. a Chen, H. Defects inspection in polycrystalline solar cells electroluminescence images using deep learning. *IEEE Access*. IEEE. 2020, sv. 8, s. 40547–40558.

[17] Ren, S., He, K., Girshick, R. a Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015, sv. 28.

[18] Ronneberger, O., Fischer, P. a Brox, T. U-net: Convolutional networks for biomedical image segmentation. In: Springer. *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. 2015, s. 234–241.

[19] Salazar, A. M. a Macabebe, E. Q. B. Hotspots detection in photovoltaic modules using infrared thermography. In: EDP Sciences. *MATEC web of conferences*. 2016, sv. 70, s. 10015.

[20] Szeliski, R. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[21] Tuğrul, K. M. Drone Technologies and Applications. In: *Drones-Various Applications*. IntechOpen, 2023. ISBN 978-0-85466-163-3.

[22] Vega Díaz, J. J., Vlaminck, M., Lefkaditis, D., Orjuela Vargas, S. A. a Luong, H. Solar panel detection within complex backgrounds using thermal images acquired by UAVs. *Sensors*. MDPI. 2020, sv. 20, č. 21, s. 6219.

[23] Venkatakrishnan, G. R., Rengaraj, R., Tamilselvi, S., Harshini, J., Sahoo, A. et al. Detection, location, and diagnosis of different faults in large solar PV system—a review. *International Journal of Low-Carbon Technologies.* 2023, sv. 18, s. 659–674. DOI: 10.1093/ijlct/ctad018. ISSN 1748-1317. Dostupné z: https://doi.org/10.1093/ijlct/ctad018.

[24] Wei, S., Li, X., Ding, S., Yang, Q. a Yan, W. Hotspots Infrared detection of photovoltaic modules based on Hough line transformation and Faster-RCNN approach. In: IEEE. *2019 6th international conference on control, decision and information technologies (CoDIT).* 2019, s. 1266–1271.

[25] Zhao, Z.-Q., Zheng, P., Xu, S.-t. a Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems.* IEEE. 2019, sv. 30, č. 11, s. 3212–3232.