

# Image retrieval: klasifikace světelného znečištění na základě porovnávání fotografií oblohy

MI-VMM.16  
2018/19

Anna Moudrá,

## Popis projektu

Výsledkem projektu je aplikace, která dovede zařadit vstupní jasovou mapu do databáze, seřazené podle znatelnosti světelného znečištění. Z reprezentativních vzorků z každého vstupního obrázku aplikace určí dominantní barevnost a porovná ji s barevnou škálou určující hodnotu světelného znečištění. Pomocí takto získaných koeficientů je vstupní obrázek zařazen do databáze.

Vstupem aplikace je jasová mapa fisheye fotografie noční oblohy.

Výstupem aplikace je databáze jasových map, seřazená od nejlepších případů po ty nejhorší.

Vedlejším výstupem aplikace jsou koeficienty (metrika) zevrubně popisující stav světelného znečištění na fotografii.

## Způsob řešení

Celý proces se skládal z několika dílčích kroků:

1. Normalizace vstupních dat.
2. Výběr zajímavých oblastí.
3. Extrakce dominantních barev ze vzorků za použití k-means shlukování barevných složek jednotlivých vzorků.
4. Analýza vybraných barev. Zahození nevhodných dat.
5. Získání koeficientů z každé zachované oblasti.
6. Porovnání koeficientů se stávající databází.

### Normalizace

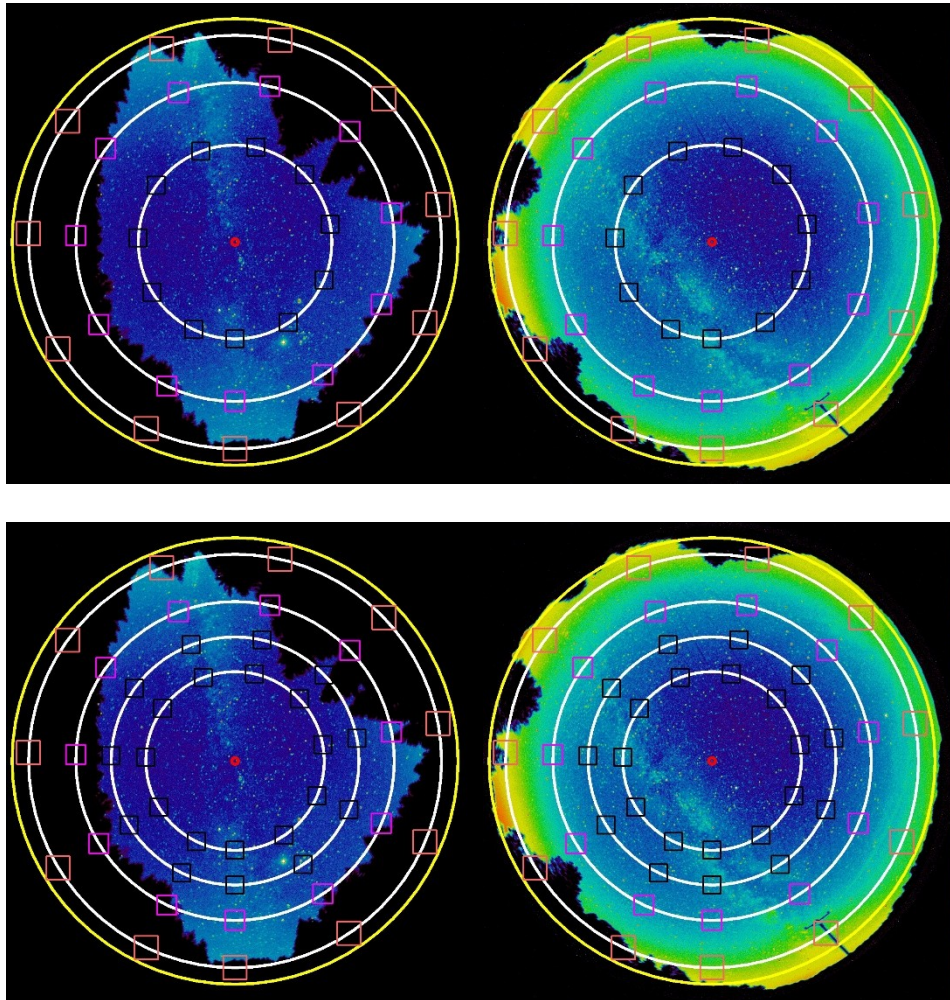
Vstupní jasové mapy je třeba přeškálovat na jednotnou velikost (1454x1444 px) tak, abychom mohli využívat konstantní parametry pro výběr vzorků.

Aplikace předpokládá, že na vstupu jsou již *jednotně zpracovaná data* (programem raw2lum nebo jiným), tak, aby barevnost i množství barevných pixelů byly srovnatelné bez dalších parametrů (typ fotoaparátu, clony atp.)

### Výběr oblastí

Jelikož sledujeme situaci v závislosti na pozorovacím úhlu a nezávisle na otočení, obrazové vzorky byly vybírány v kruzích. 10 a více vzorků na kruh se ukázalo jako ideální množství, nicméně některá data jsou velmi citlivá na počet kruhů. Zpočátku se zdálo, že 3 kruhy budou postačující, nicméně čím podobnější jsou vstupní data

tím vhodnější je zvětšit hustotu kruhů. Naopak pro některé vstupy nakonec bylo nutné hodnoty získané z krajních kruhů zcela zanedbat.



#### K-means pro výběr dominantních barev

Zprvu byla z každého vzorku vybrána pouze jedna dominantní barva. Toto se ukázalo nepraktické, jelikož výběrem většího množství barev bylo možné některé vzorky vyhodnotit jako chybové a ve výpočtu koeficientu je nepoužít.

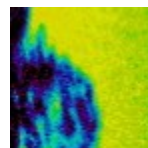
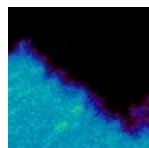
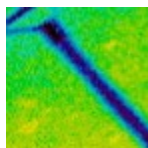
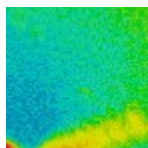
Pro každý vzorek jsou získány 2 nejpravděpodobnější se vyskytující barvy algoritmem K-means. Pro počítání vzdálenosti mezi jednotlivými barevnými vektory je použita Euklidovská metrika. Pro barvy bylo potřeba spočítat jejich vzájemný poměr ve vzorku a následně je převést na číselnou hodnotu určující míru znečištění. Jelikož program raw2lum použitý k tvorbě jasových map používá vlastní barevnou škálu, převedení barev na číselnou hodnotu spočívalo ve vyhledání nejpodobnější barvy na škále a získání její souřadnice. Pokud vzorek nebyl vyhodnocen jako chybný, výslednou číselnou hodnotou celého vzorku byl součet číselných hodnot barev získaných ze škály násobených poměrem jejich výskytu.



## Detekce chybných vzorků

Obzor se u jednotlivých lokalit liší a některé z vybraných vzorků, které se nacházejí blíže horizontu, mohou obsahovat jiné objekty než oblohu. Nejčastěji jde o stromy či infrastrukturu nalézající se v blízkosti fotografa (tmavé objekty) ale někdy se do fotografie promítne i odlesk z objektivu fotoaparátu nebo třeba rozsvícené signální světlo (bílé až červené objekty). Tyto objekty mohou nepříjemně ovlivnit vybrané dominantní barvy nebo celý vzorek znehodnotit.

Řešení tmavých okrajů je triviální, stačí nepočítat s dominantní barvou, pokud je tmavší než nějaký daný práh. V takovém případě je výsledný koeficient počítán jen z dostatečně světlé barvy, o které můžeme říci, že je pravděpodobně světlá obloha. Navíc pokud je tmavá barva zastoupena ve vzorku většinově, celý vzorek je vhodné zahodit a počítat pouze s lepšími daty. Bohužel ne všechny rušivé objekty jsou na jasové mapě dostatečně tmavé nebo velké; např. vybraný vzorek se sloupem s elektrickým vedením. Jelikož sloup je světle modrý, což je ideální hodnota neznečištěné oblohy blízko obzoru, celý vzorek nám výrazně zlepšuje stav vyhodnoceného znečištění. Vhodným řešením se ukázalo zahození vzorku, pokud se jeho dvě dominantní hodnoty (které nebyly vyhodnoceny jako „ne obloha“) nevyskytují blízko sebe na barevné škále světelného znečištění.



## Výpočet koeficientu a porovnávání vážených koeficientů

Jelikož velké množství světla blízko obzoru nemusí nutně mít vliv na viditelnost noční oblohy, aplikace dává vypočteným koeficientům z vnitřních kruhů větší váhu. Tedy i fotografie oblohy, která má po okrajích oranžové nebo žluté lemy, může být ve skutečnosti lepší než jiná, která má světla z blízkých měst skrytá za stromy.

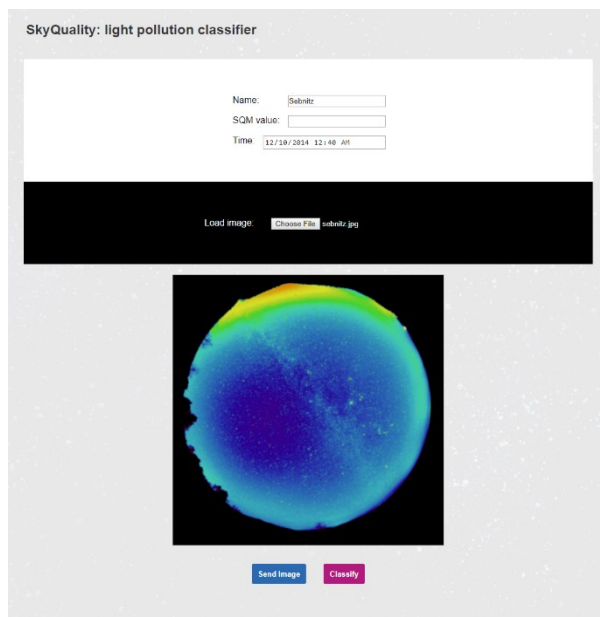
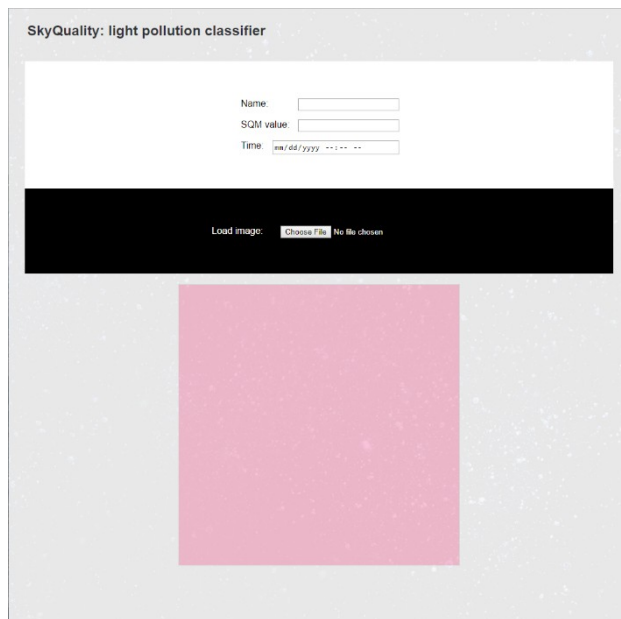
Jednotlivé koeficienty kruhů jsou uloženy tak, aby se pro každý nově přidaný obrázek nepřepočítávala celá databáze znovu.

## Implementace

Aplikace modelu klient-server je implementována v NodeJS, síťová komunikace probíhá skrze SocketIO websocketsy a obrazová data manipulují pomocí knihoven OpenCV (verze 3.4.3) a NodeJS knihovny opencv4nodejs.

Klientská část aplikace (ve složce *sqweb*) se stará o obsluhu aplikace a vykreslení výsledků. Aplikace má celkem 3 náhledy:

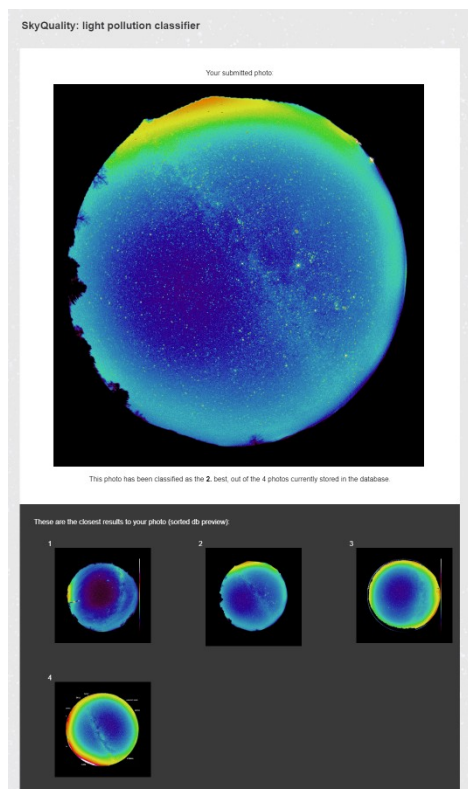
- Úvodní panel pro načtení souboru, případně dalších informací.
- Čekací stránku, která je vidět během zpracovávání dat.
- Výsledný náhled databáze s informací o indexu zařazení a náhledem 10ti nejblížešších fotek ze seřazené databáze.



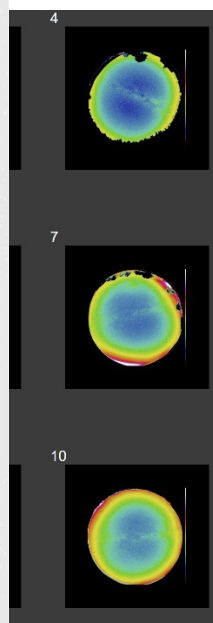
Serverová část aplikace se pak stará o samotnou klasifikaci a manipulaci se soubory.

Obsluha požadavků a síťová komunikace je implementována v souboru *app.js*, algoritmus pro výběr dominantních barev je v modulu *colorpicker.js* a veškerá zbylá manipulace s daty (výběr vzorků, porovnávání, práce s databází etc.) je v modulu *classifier.js*. Oba dva moduly jsou ve složce *my\_libs*. Složky pro ukládání dat (*my\_lib*, *result\_lib*, *samples\_lib*, *temp* a *order.txt*) jsou o adresář výše.

Data jsou posílána i ukládána ve formátu JSON, obrazová data si aplikace po síti vyměňuje v kódování Base64.

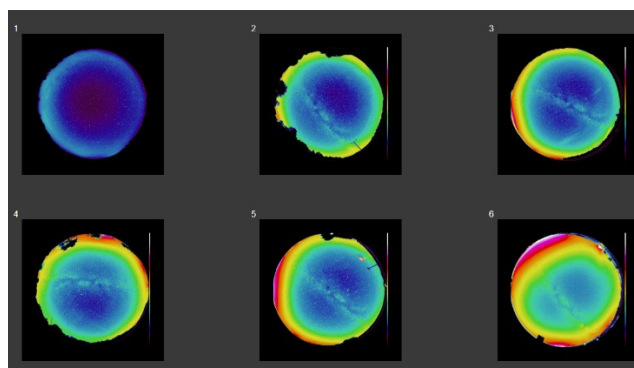


sových map může vypadat takto:



10 záznamů, zobrazí se vše.

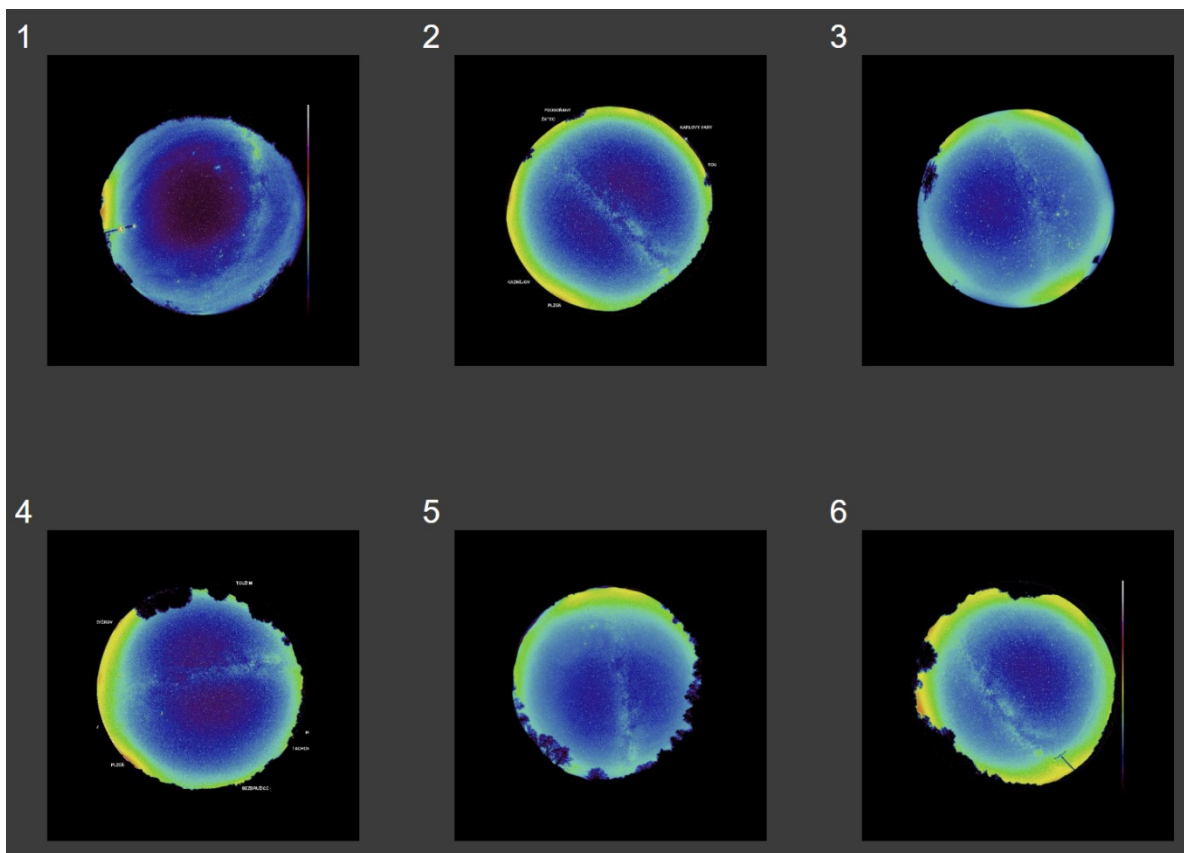
## Experimentální sekce a diskuze



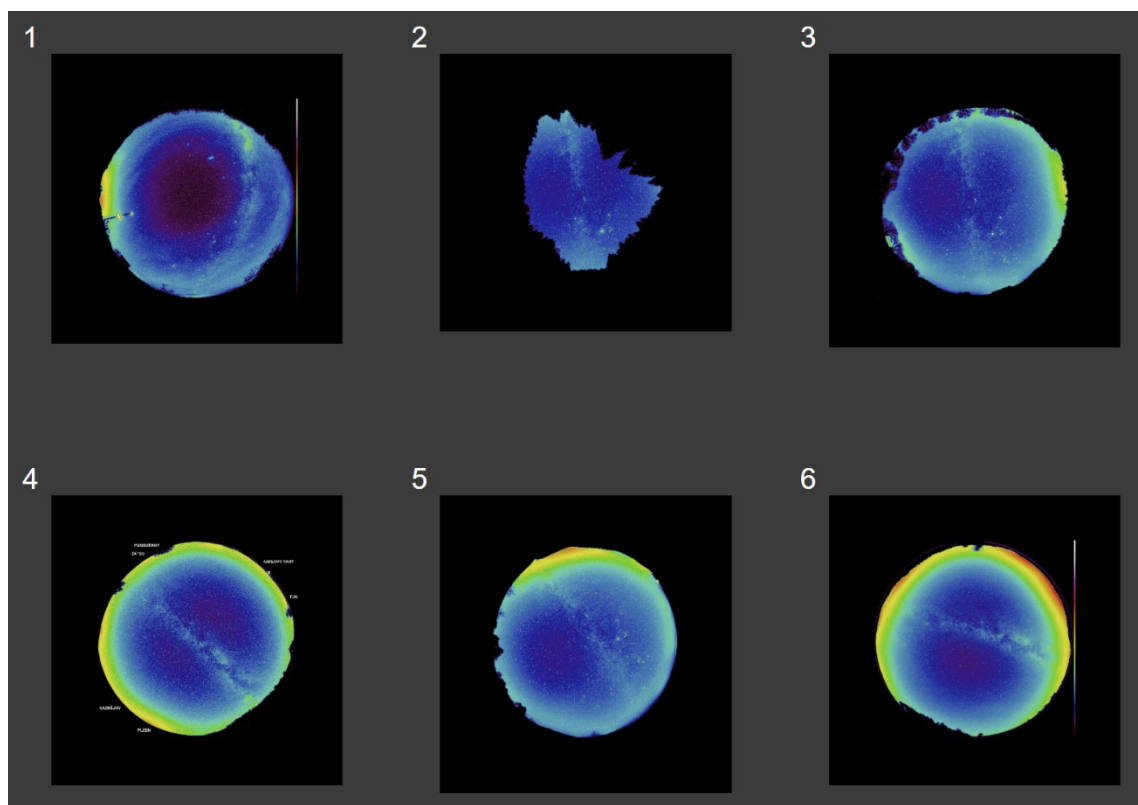
Původní vývoj začal pouze s deseti fotkami, které měly tu výhodu, že od sebe byly velmi dobře rozlišitelné a měly velmi málo rušivých okrajů. Na toto seřazení stačily jen dva kruhy s celkem 16ti vzorky, ale data jsou natolik rozdílná, že stejně dobře by pravděpodobně posloužilo i porovnávání histogramů.

S více podobnými daty ovšem začaly problémy s detekcí malých rozdílů, a proto bylo nutné zvýšit počet kruhů na 4. *Ukázka:* původní aplikace nebyla dostatečně citlivá na přechody blíž ke středu obrázku, zde by 4. obrázek měl být zařazen na 2. místě.

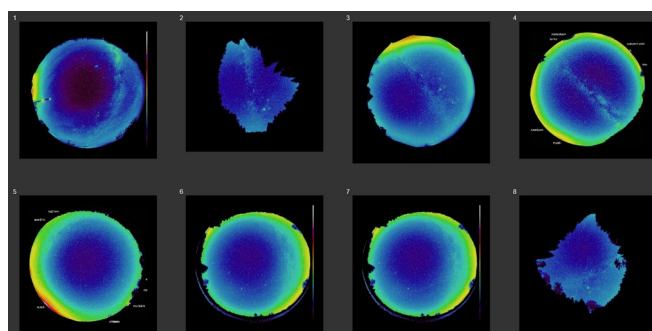




Další zajímavá testovací data byla nafocena v lese, tudíž jeden až dva vnější kruhy vzorků nejsou vůbec využity, případně je použit třeba jen 1 vzorek z 10, což není dostatečně reprezentativní. Lepší je tedy provést kontrolu, zda koeficienty vnějších kruhů nejsou podezřele špatné v porovnání s vnitřními kruhy a v případě potřeby je při porovnání nepoužít. *Ukázka:* Druhý obrázek byl řazen jen pomocí porovnávání koeficientů z vnitřních dvou kruhů. Při použití všech kruhů byl nesprávně řazen skoro až na úplný konec databáze. To z toho důvodu, že okraje stromů bývají na takto zpracovaných fotkách fialové a z krajních kruhů se vybralo jen velmi málo vzorků. Tyto vzorky se zprůměrovaly na hodnotu víc růžovou než tmavě modrou, což je kritická hodnota. Tmavě modré barvy, které by tuto hodnotu opět vylepšily, neprošly přes zadaný prah jasu pro vnější kruh (suma RGB hodnot dominantní barvy menší než 270).



Bohužel ne na všechna data je vhodné jedno nastavení parametrů, a proto by bylo pro aplikaci vhodnější, aby si sám uživatel nastavil okraje oblohy tak, aby výsledky byly spolehlivější. Pro použití v praxi by se také určitě hodila detekce duplikátů.<sup>1</sup>



Ze začátku nebylo příliš jasné, jak velký vliv bude mít pozice Mléčné dráhy na výběr dominantních barev, a jestli nebude nutné ji detekovat a případně ji odmazat. Nicméně zdá se, že díky výběru příznaků v kruzích je klasifikátor odolný i na její různé pozice a klasifikuje správně, dokonce i když není Mléčná dráha vidět.

## Závěr

Přestože implementace je poměrně přímočará, aplikace vrací lepší výsledky, než bylo původní očekávání.

<sup>1</sup> Do té doby budiž útechou, že se duplikáty zařazují správně k sobě.

Jelikož se jednalo o experiment, kde nebylo dopředu jasné, jak dobrých výsledků lze dosáhnout a zda má tento nápad využití v praxi (pro amatérské astronomy a ekology), je velmi dobrým výsledkem, že je možné na základě fotografií poměrně přesně porovnávat a řadit jednotlivé lokality podle jasů noční oblohy. Oproti dosavadním metodám (porovnávání hodnot naměřených s přístrojem Sky Quality Meter (SQM) nebo výstupů z programu raw2lum) je mimořádnou výhodou, že klasifikátor správně bere v úvahu jas oblohy u obzoru. SQM měří pouze úhel 60 stupňů u zenitu, přitom jas u obzoru je klíčovou složkou celkové klasifikace lokality.

Nespornou nevýhodou současného stavu aplikace je závislost na externím softwaru pro tvorbu jasových map z fotek noční oblohy, což vede k omezené využitelnosti.