

# NI-VMM: Podobnost obrázků (SIFT)

Igor Súlovský | David Mikulka

sulovigo@fit.cvut.cz

mikulda3@fit.cvut.cz

ČVUT FIT - Praha — December 11, 2022

## Popis projektu

Projekt se zaměřuje na implementaci porovnávání obrázků na základě jejich SIFT deskriptorů. Po nahrání obrázku uživatelem webová aplikace zobrazí podobné obrázky z databáze seřazené dle podobnosti. Uživatel si také může zvolit počet deskriptorů, které se budou porovnávat. Tento počet má následně dopad na to, jak rychle aplikace daný dotaz vyhodnotí.

## 1 Způsob řešení

Při spuštění vyhledávání uživatelem se nejdříve spustí generace SIFT deskriptorů pro obrázky v databázi, počet deskriptorů je zadán uživatelem ve webové aplikaci. Následně se vygeneruje stejný počet SIFT deskriptorů pro obrázek, který je také nahrán uživatelem ve webové aplikaci. Poté se deskriptory porovnávají pomocí Euklidovské vzdálenosti. Deskriptor se počítá za správný, pokud výsledná vzdálenost nepřesahuje číslo zadané na vstupu uživatelem. Po skončení výpočtu se uživateli zobrazí podobné obrázky z databáze seřazené od nejpodobnějšího po nejméně podobný.

## 2 Implementace

Pro implementaci byla zvolena knihovna *OpenCV*<sup>1</sup>, která obsahuje, mimo jiné, také algoritmus pro extrakci SIFT deskriptorů a je podporována ve více programovacích jazycích. Pro tento projekt byl zvolen jazyk Python, a to kvůli následné implementaci webového rozhraní, pro které byl použit framework *Django*<sup>2</sup> s kterým mají členové týmu zkušenosti. Dále byla také pro implementaci využita knihovna *NumPy*<sup>3</sup> kvůli zjednodušení práce se soubory a zrychlení počítání euklidovské vzdálenosti mezi deskriptory. Webové rozhraní je napsáno v HTML a stylování je zařízeno za využití knihovny *Bootstrap*<sup>4</sup>.

## 3 Instalace a spuštění

Pro spuštění aplikace je třeba nainstalovat Python (verze 3.9 a vyšší) a doplnit jej o knihovny *OpenCV*, *Numpy* a také nainstalovat framework *Django*. Pro rozšíření stávající databáze o vlastní obrázky stačí do složky `/static/images/img` přidat libovolné množství obrázků s příponou `.jpg`

Po instalaci všech potřebných závislostí je možné webovou aplikaci spustit v lokální instanci a to pomocí příkazu `./manage.py runserver` (pokud se nacházíme ve složce se souborem `manage.py`). Po spuštění programu otevřete libovolný webový prohlížeč a aplikaci najdete na adrese `localhost:8000`. Pokud vám to nebude fungovat, je možné, že je port obsazen jinou službou a stačí vybrat jiný port pomocí `./manage.py runserver <libovolný port>`.

Webová aplikace obsahuje pole pro nahrání obrázku, který se zobrazí vedle v náhledovém boxu. Uživatel si poté může na slideru nastavit míru podobnosti v rozmezí hodnot 1 až 9. Druhou položkou, kterou

---

<sup>1</sup><https://docs.opencv.org/4.x/index.html>

<sup>2</sup><https://www.djangoproject.com/>

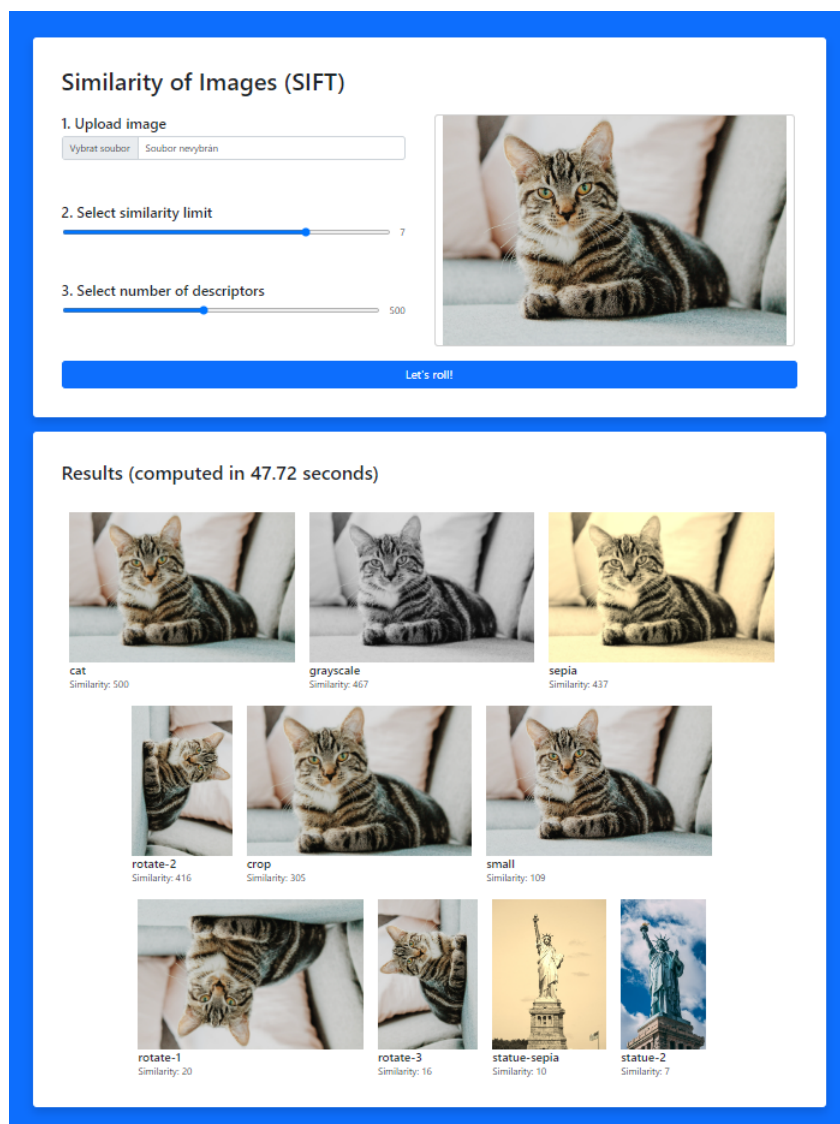
<sup>3</sup><https://numpy.org/>

<sup>4</sup><https://getbootstrap.com/>

si uživatel může nastavit, je počet deskriptorů pro výpočet, kde lze nastavit hodnotu v rozmezí 100 až 1000 (se skokem o 100). Po kliknutí na tlačítko *Let's roll!* začne výpočet. Výsledky se zobrazí ve spodním okně, kde se v závorce zároveň zobrazí čas s dobou výpočtu.

## 4 Příklad výstupu

Celkový vzhled aplikace a zároveň příklad výstupu pro míru podobnosti rovnou 7 a počet deskriptorů roven 500 si lze prohlédnout na obrázku 1. Uživatelské rozhraní je intuitivní.



Obrázek 1: Screenshot aplikace

## 5 Experimentální sekce

V prvním experimentu testujeme vliv počtu porovnávajících se deskriptorů na dobu výpočtu. Jak můžeme vidět v tabulce 1, nižší počet deskriptorů nám urychluje celkový výpočet a to až v desítkách sekund. Při snížení počtu deskriptorů také snižujeme přesnost, ale na vybraných obrázcích to však nebylo signifikantní.

V druhém experimentu testujeme vliv zvolené hranice podobnosti na výsledné seřazení. Pro experiment byl vybrán obrázek kočky a stejný obrázek, na který byl aplikován sepia filter, jak lze vidět na obrázku 2. Měřili jsme počet správně vybraných párů při různě zvolené hranici podobnosti. Čím vyšší tato hranice

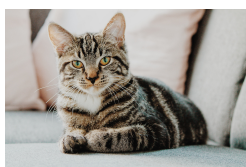
Počet deskriptorů	Počet obrázků v databázi	Čas (s)
500	10	47.72
	22	56.98
700	10	75.32
	22	85.05
1000	10	142.12
	22	160.38

Tabulka 1: Experiment zaměřen na počet deskriptorů

byla, čím větší množství podobností bylo nalezeno, jak lze vidět v tabulce 2. Počet testovaných deskriptorů při testování byl 200. Při testování jsme si také všimli, že u vyšší hranice se výrazně zvyšovala podobnost i u obrázků, které si podobné nebyly.

Hranice podobnosti	Počet podobných deskriptorů
2	156
5	175
9	180

Tabulka 2: Experiment zaměřen na vliv změny hranice podobnosti



(a) Obrázek kočky



(b) Obrázek kočky se sepia filtrem

Obrázek 2: Obrázky pro porovnávání

V posledním experimentu zkusíme, jak různé změny obrázku (ne)zachovávají podobnost s originálem. Pro tento experiment jsme využili obrázek z předchozího experimentu a dva další, které jsme upravili stejným způsobem, jako první obrázek. Pro úpravu obrázků jsme použili program Gimp. Zajímalo nás, jak se algoritmus bude chovat, když obrázek různě otočíme, změníme paletu barev a nebo zmenšíme jeho velikost. Pro experiment jsme použili 500 deskriptorů. Ve většině případů se algoritmus zachoval tak, jak bychom očekávali, což můžeme vidět v tabulce 3. Naměřené hodnoty jsou počet podobných deskriptorů. Překvapila nás míra podobnosti u rotace, na kterou byl ještě použit efekt zrcadlení. V případě zrcadlení a otočení o 90° a 180°, byla míra podobnosti velmi nízká. Algoritmus také nezvládl úpravu fotek pomocí cartoon efektu. Všechny použité obrázky je možné najít ve složce `/static/images/img`

Změna v obrázku	Kočka	Socha svobody	Socha Davida
Grayscale	467	447	466
Sepia	437	377	457
Rotace o 90°	16	49	42
Rotace o 180° + zrcadlení	20	52	49
Rotace o 270° + zrcadlení	416	411	432
Ořezání	305	271	411
Zmenšení	109	112	213
Cartoon	1	17	24

Tabulka 3: Experiment zaměřen na podobnost při úpravě obrázku

## 6 Diskuze

Aplikace je díky knihovně Bootstrap responzivní pro libovolně zvolené rozlišení. Aplikace také uživateli umožňuje zvolit si počet deskriptorů, které se budou porovnávat a podobnostní mírou, která se využívá při výběru korektní shody. Výběrem počtu deskriptorů může uživatel výrazně ovlivnit délku běhu. Po výpočtu se zobrazí všechny takové obrázky z databáze, které mají míru podobnosti větší než 0, tedy se nezobrazí takové obrázky, u kterých žádný porovnávaný deskriptor nebyl napárován.

Nedostatek aplikace vidíme v nemožnosti zvolit si jinou metodu na výpočet (např. Signature Quadratic Form Distance), nicméně program je pro implementaci této metody připraven.

## Závěr

Výstupem našeho snažení je funkční webová aplikace pro vyhledávání obrázků na základě podobnosti s využitím SIFT deskriptorů s jednoduchým, ale přehledným a responzivním webovým rozhraním. Aplikace funguje pro libovolný nahraný obrázek a při zvolení vyššího počtu deskriptorů umožňuje (za cenu delšího času) přesnější výsledky.