



RAZVOJ NPM PAKETA ZA EKSTRAKCIJU NAJFREKVENTNIJIH BOJA SA SLIKE DEVELOPMENT OF NPM PACKAGE FOR EXTRACTING MOST FREQUENT COLORS

Jelena Bojanić, Milan Vidaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – Tema ovog rada jeste razvoj *npm* (*node package manager*) paketa (biblioteke) za rad sa bojama i izdvajanje najfrekventnijih boja sa slike.

Ključne reči: *npm* paket, obrada slike, *TypeScript*, *JavaScript*, *HSL*, *RGB*

Abstract – *This paper deals with the development of an npm package for working with colors and extracting the most frequent colors from an image.*

Keywords: *npm* package, image processing, *TypeScript*, *JavaScript*, *HSL*, *RGB*

1. UVOD

Kako bi se boje tačno izdvojile sa slike potrebno je izvršiti detaljnu analizu svih piksela, i na adekvatan način obraditi prikupljene informacije. Korišćenje gotovih rešenja u tom slučaju izuzetno je korisno, jer razvoj takvih rešenja zahteva visok nivo domenskog znanja (autor takvog rešenja mora dobro poznavati boje, razlike između boja i njihovih nijansi). Ovaj rad se bavi kreiranjem *npm* (*node package manager*) paketa *color-catcher* koji će omogućiti obradu slike i ekstrakciju najfrekventnijih boja koje se na toj slici nalaze. Posmatrani paket pruža i dodatne funkcionalnosti za rad sa bojama, obezbeđujući mogućnost dobijanja palete boja koje su u harmoniji sa predstavljenom bojom, na osnovu korisničkog unosa. Harmonije boja koje paket podržava su komplementarna, monohromatska, triadska i tetradska harmonija.

2. PREGLED SLIČNIH SISTEMA

Pre detaljnijeg navođenja funkcionalnosti i implementacije opisanog sistema, biće izvršena analiza sličnih sistema. Sva posmatrana rešenja biće isključivo biblioteke koje su dostupne kao paketi u *npm* registru. Da bi se neko rešenje smatralo sličnim, potrebno je da ono vrši analizu slike i identifikaciju boja koje se na toj slici nalaze. Jedan od takvih sistema je *Extract Colors* [1]. Dato rešenje predstavlja *TypeScript* biblioteku otvorenog koda, koju je razvila *node.js* zajednica. Može se koristiti u klijentskom i serverskom delu veb aplikacije.

Pružila opciju podešavanja velikog broja parametara, a neki od njih su količina piksela, razlika između sličnih boja i razlika u zasićenosti. Ono što je izdvaja od drugih, jeste i

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Milan Vidaković, red. prof.

činjenica da se za svaku pronađenu boju na slici takođe dobija informacija o zastupljenosti date boje ili njoj sličnih boja.

Jedna od poznatijih biblioteka za obradu slika u *npm* zajednici jeste *Color Thief* [2]. Napisana u *JavaScript* programskom jeziku, a obradu boja sa slike vrši putem procesa kvantizacije boja. Kvantizacija boja je proces koji uključuje prevođenje velikog broja boja u manji skup boja. Zbog upotrebe ove tehnike, posmatrano rešenje sadrži određen stepen greške, koji u zavisnosti od upotrebe može biti od manjeg ili većeg značaja. Nudi funkciju za izdvajanje dominantne boje sa slike, kao i funkciju za izdvajanje palete boja sa slike. Funkcionalnost koja bi učinila ovo rešenje boljim bi svakako bila informacija o tome koliko je posmatrana boja zastupljena na slici.

3. OPIS KORIŠĆENIH TEHNOLOGIJA

Rešenje je napisano u *TypeScript* [3] programskom jeziku. *TypeScript* je nadskup *JavaScript*-a [4], što znači da je svaki važeći *JavaScript* kod takođe i važeći *TypeScript* kod. Paket je objavljen pomoću *npm* centralnog repozitorijuma za distribuciju *JavaScript* paketa. Koristi *Canvas* API [5] koji obezbeđuje manipulaciju i rad sa grafikom unutar veb pregledača. Za rad sa bojama, koristi *RGB* [6] i *HSL* [7] modele boja. Za detekciju razlika između boja, upotrebljena je *color-diff* [8] biblioteka koja računa razliku između dve boje upotrebom *CIEDE2000* algoritma.

3.1 TypeScript

TypeScript je programski jezik otvorenog koda, koji razvija i održava *Microsoft*. Predstavlja nadskup *JavaScript*-a i dodaje jeziku opciono statičko i dinamičko tipiziranje. Omogućava definisanje tipova podataka i pruža napredne funkcionalnosti kao što su interfejsi, generički tipovi i enumeracija. Olakšava održavanje i povećava bezbednost tokom razvoja. Integracijom sa popularnim alatima za razvoj, postaje snažan izbor za izgradnju skalabilnih i održivih aplikacija.

3.2 .NPM (Node Package Manager)

Node Package Manager je zvanični centralni repozitorijum *node* modula, kao i softver za upravljanje bibliotekama od kojih aplikacija zavisi. Koristi se prvenstveno za upravljanje zavisnostima u *node.js* projektima, ali se takođe koristi za upravljanje paketima u *JavaScript* projektom okruženju. Sastoji se od konzole i baze podataka javnih i privatnih paketa, koji se još nazivaju i *npm* registri. *Npm* pruža i opciju kreiranja i objavljivanja vlastitih paketa odnosno biblioteka. Samo-

objavljivanje paketa na npm registru omogućava širenje znanja, efikasnost u razvoju, brzu integraciju gotovih rešenja, održavanje standarda, što doprinosi dinamičnom i inkluzivnom okruženju za razvoj softvera.

3.3. Canvas API

Canvas API je deo HTML5 [9] standarda i predstavlja tehnologiju koja omogućava dinamičko crtanje i rad sa grafikom koristeći *JavaScript*. Između ostalog, može se koristiti za manipulaciju pikselima u realnom vremenu. Od posebnog značaja za posmatrano rešenje jeste *canvas* element i *canvas* kontekst.

Element *canvas*-a je stvarni DOM [10] čvor koji je ugrađen u HTML stranicu. Kontekst *canvas*-a je objekat sa svojstvima i metodama koje se mogu koristiti za prikazivanje grafike unutar *canvas* elementa. Kontekst može biti 2D ili 3D. Svaki element koji je tipa *canvas*, može imati samo jedan kontekst.

3.4. Color-diff

Color-diff je *JavaScript* biblioteka koja, upotrebom CIEDE2000 [11] algoritma, pruža funkcije za rad sa bojama i njihovo poređenje.

International Commission on Illumination (CIE) [11] je organizacija koja se bavi standardizacijom u oblastima kolorimetrije, fotometrije i drugih srodnih disciplina. Formirana je početkom 1900-ih godina kako bi razvila standarde i preporuke za date oblasti i kako bi se omogućila konzistentnost i međunarodno razumevanje u proučavanju svetlosti, boja i njihovih efekata. Odigrala je ključnu ulogu u oblasti poređenja boja, predstavljajući svetu koncept Delta E.

Delta E je mera promene u vizuelnom opažanju dve date boje. Ova metrika se koristi za kvalifikaciju razlike u boji na onaj način kako je percipira ljudsko oko. Izraz delta potiče iz matematike i označava promenu. Dodatak E odnosi se na nemačku reč *Empfindung*, što se prevodi kao senzacija.

Delta E vrednost se računa na osnovu razlika u parametrima boje kao što su nijanse, zasićenje i svetlina. Na tipičnoj skali, vrednost se kreće između 0 i 100:

- ≤ 1.0 ne vidi se ljudskim okom,
- 1-2 vidi se detaljnim posmatranjem,
- 2-10 vidi se na prvi pogled
- 11-49 boje su više slične nego suprotne i
- 50-100 boje su suprotne.

4. SPECIFIKACIJA

Zadatak obuhvata izradu npm paketa, čija glavna funkcionalnost predstavlja izdvajanje najfrekventnijih boja sa slike.

4.1. Funkcionalni zahtevi

Prilikom razvoja npm paketa, funkcionalni zahtevi obuhvataju definisanje jasnog API-ja i parametara funkcije, izradu dokumentacije, obradu grešaka, kao i osiguravanje konzistentnosti u radu paketa. Budući da postoje brojni slučajevi korišćenja, postoji nekoliko parametara koje je moguće definisati prilikom pozivanja funkcije za ekstrakciju boja sa slike. Korisnicima je

omogućeno da utiču na preciznost i brzinu procesa promenom *blurFactor* parametra, koji određuje koliko se susednih piksela "smešta" u rezultujući piksel.

Broj najfrekventnijih boja koje korisnik želi da dobije zavisi od specifičnosti upotrebe, stoga je važno definisati parametar koji omogućava preciziranje željenog broja boja. Jedan od važnih funkcionalnih zahteva jeste dobro definisan API, što se izražava u dobro definisanim ulaznim i izlaznim parametrima funkcije. Ulazni parametri su tipizirani, a biblioteka je implementirana na način da adekvatno upravlja greškama. Prilikom unosa neispravnih tipova podataka, korisnik dobija poruku sa opisom detektovane greške.

Izlazna vrednost funkcije predstavlja niz boja, koje su sortirane u opadajućem redosledu prema procentu zastupljenosti svake boje na slici. Svaki element u rezultujućem nizu sadrži vrednost boje u RGB i HEX formatu, kao i zastupljenost date boje u procentima.

Idealno je da ovakav vid rešenja obuhvati što veći spektar slučajeva upotrebe, a poželjna je i pismena dokumentacija, koja je jednako bitna kao i sama izrada i implementacija. Dokumentaciji posmatrane biblioteke moguće je pristupiti putem npm veb sajta, gde su jasno definisane sve njene funkcionalnosti, način instalacije, način korišćenja kao i tipovi ulaznih i izlaznih parametara.

4.2. Nefunkcionalni zahtevi

Sa obzirom na primarnu upotrebu ovakvih biblioteka u klijentskim aplikacijama, neophodno je postići minimalno vreme odziva i brzu instalaciju. Ovakvi aspekti su važni radi postizanja optimalne efikasnosti u radu biblioteke. Prosečno vreme odziva posmatranog rešenja je 100ms, što ujedno predstavlja i minimalno prihvatljivo vreme koje se očekuje na klijentskoj strani.

5. IMPLEMENTACIJA

Opisana biblioteka sastoji se od pet funkcija. Četiri funkcije obezbeđuju rad sa harmonijama boja dok se ključna funkcionalnost biblioteke bavi izdvajanjem najfrekventnijih boja obradom slike. Biblioteka je napisana u *TypeScript* programskom jeziku.

5.1 Ekstrakcija najfrekventnijih boja

Predstavlja ključnu funkcionalnost posmatrane biblioteke. Ulazni parametar je objekat koji sadrži polje *imageURL*, odnosno putanju do učitane slike koju treba obraditi, polje sa nazivom *n* odnosno dužinu rezultujućeg niza kao i *blurFactor* koji označava koliko će se susednih piksela sjediniti u jedan prilikom zamućenja ulazne slike. Struktura funkcije može se podeliti u tri celine:

- obrada ulazne slike,
- klasifikacija i
- analiza boja.

Obrada slike podrazumeva učitavanje ulazne slike, proporcionalno smanjenje njenih dimenzija i iscrtavanje na *canvas* element. *Canvas* API omogućava pristup piksel podacima slike. Dužina niza koji predstavlja piksel podatke direktno zavisi od dimenzija slike. Slike visoke rezolucije mogu sadržati i milione piksel podataka koji ih

opisuju. Zbog toga je korak proporcionalnog smanjenja dimenzija slike od izuzetne važnosti, kako bi se umanjio ukupan broj piksela koji je potrebno obraditi bez gubitka informacija.

Iz konteksta *canvas* elementa pristupa se nizu koji predstavlja piksel podatke date slike, čime se započinje sledeća celina, odnosno klasifikacija boja. Klasifikacija boja je proces koji obuhvata iteraciju kroz piksele slike, identifikaciju boje na svakom pikselu i kategorizaciju te boje u odgovarajuću klasu na osnovu definisanih karakteristika.

Upotrebom HSL modela, boje mogu klasifikovati u grupe. Pogodan kriterijum klasifikacije, jeste nijansa odnosno ugao na kom se boja unutar kruga boja nalazi. Iz tog razloga, u posmatranom rešenju definisano je sedam klasa, gde je svaka klasa opisana nazivom, početnom vrednošću ugla od kog boje te klase počinju kao i vrednošću ugla sa kojim se boje te klase završavaju.

Nakon što je klasa za datu boju poznata, postoje tri slučaja:

- boja je već evidentirana unutar klase,
- postoji dovoljno slična boja datoj boji koja je već evidentirana unutar klase i
- boja nije evidentirana unutar klase i ne postoji dovoljno slična boja unutar iste klase koja je evidentirana.

U situaciji gde je data boja već zabeležena unutar određene klase, povećava se broj njenih ponavljanja. Ukoliko data boja nije prethodno detektovana unutar klase, neophodno je proveriti da li postoji dovoljno slična prethodno detektovana boja unutar iste klase. Za poređenje dve boje koristi se funkcija *diff* iz biblioteke *color-diff*. Proces poređenja boja se oslanja na iteraciju kroz sve boje unutar jedne klase i potragu za razlikama u odnosu na trenutnu boju.

Granična vrednost razlike između boja u ovom radu iznosi 17. Ukoliko rezultat poređenja dve boje iznosi manje od 17, smatra se da je razlika između njih toliko mala da ljudsko oko to ne uočava.

Ukoliko nijedna prethodno evidentirana boja sa sličnošću manjom od 17 nije prisutna, zaključuje se da se radi o prvom pojavljivanju posmatrane boje i da ne postoji dovoljno slična boja, već zabeležena u sistemu. Data boja se evidentira unutar detektovane klase.

Finalni korak uključuje preuzimanje boja iz klasa gde postoji barem jedna detektovana boja na slici i njihovo sortiranje prema broju ponavljanja. Za svaku prepoznatu boju izdvajaju se informacije o njenoj RGB vrednosti, HEX kodu, broju ponavljanja i procentu koji zauzima na slici. Procenat učešća boje na slici se računa kao odnos broja ponavljanja boje i ukupnog broja piksela koji su bili obrađeni.

6. OBJAVLJIVANJE BIBLIOTEKE

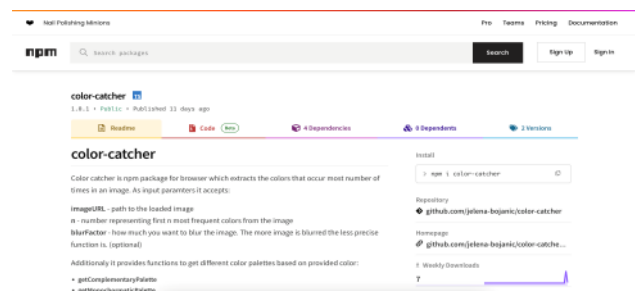
Npm pruža i opciju kreiranja i objavljivanja vlastitih paketa odnosno biblioteka. Samo-objavljivanje paketa na npm registru omogućava širenje znanja, efikasnost u razvoju, brzu integraciju gotovih rešenja, održavanje

standarda, što doprinosi dinamičnom i inkluzivnom okruženju za razvoj softvera.

Kako bi se paket kreirao potrebno je uspostaviti nalog na npmjs.com. Inicijalizacija paketa vrši se komandom *npm init*, nakon koje sledi nekoliko upita o imenu paketa, verziji paketa kao i ulaznom fajlu. Ime paketa mora biti jedinstveno i može sadržati samo mala slova.

Nakon implementacije rešenja, testiranje paketa moguće je vršiti upotrebom komande *npm link <naziv paketa>*. Ova komanda kreira *symbolic link*, odnosno vezu između lokalnog razvojnog paketa sa globalnom npm instalacijom.

Kada je rad na paketu završen, potrebno je prijaviti se na npm registar sa prethodno kreiranim nalogom putem komande *npm login*. Objavljivanje paketa realizuje se komandom *npm publish*. Svaki objavljen paket ima dodeljenu veb stranicu kao što je prikazano na slici 1. Svako naredno objavljivanje zahteva povećanje verzije paketa, a preporučuje se korišćenje semantičkog verzioniranja.



Slika 1. veb stranica objavljenog npm paketa

7. ZAKLJUČAK

Primarni zadatak ovog rada bio je razvoj npm paketa za rad sa bojama, fokusiranog na ekstrakciju najfrekventnijih boja sa slike. Osnovna ideja bila je stvaranje gotovog rešenja, usmerenog na rešavanje izazova gde je značajnija frekventnost boja nego njihova dominantnost.

Centralna funkcionalnost biblioteke jeste izdvajanje boja sa najvećim brojem ponavljanja na slici. Takođe pruža funkcije za generisanje harmonija boja na osnovu zadate boje.

Biblioteka je razvijena u trenutno savremenim tehnologijama koje se proširuju i napreduju, te ima dobru osnovu za dalji razvoj. Dodavanje opcije obrade video snimaka i primena biblioteke unutar serverskog dela aplikacije samo su neke od ideja koje bi ovu biblioteku unapredile i napravile ozbiljnu konkurenciju postojećim rešenjima.

8. LITERATURA

- [1] <https://www.npmjs.com/package/extract-colors> (pristupljeno u decembru 2023.)
- [2] <https://www.npmjs.com/package/color-thief> (pristupljeno u decembru 2023.)
- [3] <https://www.typescriptlang.org/> (pristupljeno u decembru 2023.)
- [4] <https://en.wikipedia.org/wiki/JavaScript> (pristupljeno u decembru 2023.)

- [5] https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API (pristupljeno u decembru 2023.)
- [6] https://en.wikipedia.org/wiki/RGB_color_model (pristupljeno u decembru 2023.)
- [7] https://en.wikipedia.org/wiki/HSL_and_HSV (pristupljeno u decembru 2023.)
- [8] <https://www.npmjs.com/package/color-diff> (pristupljeno u decembru 2023.)
- [9] <https://developer.mozilla.org/en-US/docs/Glossary/HTML5> (pristupljeno u decembru 2023.)
- [10] https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model (pristupljeno u decembru 2023.)
- [11] <http://cie.co.at/> CIE (pristupljeno u decembru 2023.)

Kratka biografija:



Jelena Bojanić rođena je u Novom Sadu 1997. god. Master rad na Fakultetu tehničkih nauka iz oblasti Elektrotehnike i računarstva – Računarstvo i automatika, odbranila je 2023.god.
kontakt:
jelena.bojanic97@gmail.com



Milan Vidaković je završio doktorske studije 2003. godine na Fakultetu tehničkih nauka u Novom Sadu. Na istom fakultetu je 2014. godine izabran za redovnog profesora iz oblasti Primenjene računarske nauke i informatika.