# Mjerenje parametara vina primjenom računalnog vida

**Jazvić, Ilija**



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Sveučilišni studij**

# MJERENJE PARAMETARA VINA PRIMJENOM RAČUNALNOG VIDA

**Završni rad**

**Ilija Jazvić**

**Osijek, 2023.**

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

**Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

**Osijek, 09.07.2023.**

**Odboru za završne i diplomske ispite**

# Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

| | |
|---|---|
| **Ime i prezime Pristupnika:** | Ilija Jazvić |
| **Studij, smjer:** | Računalno inženjerstvo |
| **Mat. br. Pristupnika, godina upisa:** | R4502, 27.07.2020. |
| **OIB Pristupnika:** | 73208223977 |
| **Mentor:** | izv. prof. dr. sc. Ivan Aleksi |
| **Sumentor:** | izv. prof. dr. sc. Tomislav Matić |
| **Sumentor iz tvrtke:** | |
| **Naslov završnog rada:** | Mjerenje parametara vina primjenom računalnog vida |
| **Znanstvena grana rada:** | **Arhitektura računalnih sustava (zn. polje računarstvo)** |
| **Zadatak završnog rad:** | U ovom završnom radu potrebno je implementirati metode računalnog vida za određivanje boje vina te vrijednosti pH. Boju vina potrebno je mjeriti kroz protočnu kivetu. Vrijednost pH je potrebno mjeriti određivanjem boje na papirnatoj indikatorskoj trakici koja mjenja boju u ovisnosti o pH. U radu je potrebno koristiti računalni modul s kamerom. |
| **Prijedlog ocjene završnog rada:** | Izvrstan (5) |
| **Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:** | Primjena znanja stečenih na fakultetu: 3 bod/boda<br>Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda<br>Jasnoća pismenog izražavanja: 3 bod/boda<br>Razina samostalnosti: 3 razina |
| **Datum prijedloga ocjene od strane mentora:** | 09.07.2023. |
| **Datum potvrde ocjene od strane Odbora:** | 12.07.2023. |

| | |
|---|---|
| Potvrda mentora o predaji konačne verzije rada: | Mentor elektronički potpisao predaju konačne verzije. |
| | Datum: |

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

# IZJAVA O ORIGINALNOSTI RADA

**Osijek, 17.07.2023.**

| | |
|---|---|
| **Ime i prezime studenta:** | Ilija Jazvić |
| **Studij**: | Računalno inženjerstvo |
| **Mat. br. studenta, godina upisa:** | R4502, 27.07.2020. |
| **Turnitin podudaranje [%]:** | 15 |

Ovom izjavom izjavljujem da je rad pod nazivom**: Mjerenje parametara vina primjenom računalnog vida**

izrađen pod vodstvom mentora izv. prof. dr. sc. Ivan Aleksi

i sumentora izv. prof. dr. sc. Tomislav Matić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

# SADRŽAJ

# 1. UVOD

U današnjoj proizvodnji vina, kiselost vina je ključna za određivanje ukusa, izgleda i izrade vina te ju je zbog toga potrebno pratiti. Osnovni način i najjednostavniji način za mjerenje kiselost vina je pomoću pH skale, gdje manji broj pH označava na višu razinu kiselosti, a veći pH broj označava nižu razinu kiselosti. Vrijednost pH se mjeri određivanjem boje na papirnatoj indikatorskoj trakici odnosno lakmus papiru koji, mijenja boju u ovisnosti o pH gdje boje idu od crvene do plave odnosno ljubičaste boje, gdje je crvena jako kiselo dok plava i ljubičasta boja označava lužnatu tekućinu.

Mikroupravljač koji je korišten tijekom izrade ovog završnog rada je *ESP32-CAM*. Sučelje koje je korišteno za programiranje *ESP-a* je *Arduino* integrirano razvojno okruženje *(*engl. *IDE).* *ESP32-CAM* je korišten zbog njegove jednostavne implementacije i niske cijene, taj modul ima ugrađenu kameru koja služi za dobivanje informacija o kiselosti vina, te je ovaj modul savršen za rješavanje problema mjerenja pH vrijednosti vina.

U drugom poglavlju je objašnjena teorijska pozadina u vinskoj industriji i važni parametri vina, zašto je kiselost vina važan faktor za praćenje u vinarijama, pregled tradicionalnih metoda mjerenja kiselosti vina te ostale metode koje postoje za obavljanje istog zadatka. U trećem poglavlju su detaljno opisane tehnologije korištene za rješavanje pomenutog problema. U četvrtom poglavlju je opisan eksperimentalni postupak, detaljno opisane metode koje su korištene za mjerenje parametara vina, objašnjeno je kako su prikupljeni podaci i kako su obrađeni. U petom poglavlju je raspravljano o rezultatima mjerenja sustava i objašnjena je učinkovitost sustava. U zaključku je prokomentirano rješenje sustava za mjerenje pH vrijednosti vina te koje su potrebne izmjene odnosno unaprjeđenja za daljnje usavršavanje čitavog sustava.

## 1.1. Zadatak završnog rada

U završnom radu potrebno je izraditi funkcionalan sustav za mjerenje kiselosti vina uz pomoć lakmus papira i računalnog vida koji će biti u nastavku opisan.

# 2. TEORIJSKI PREGLED

## 2.1. Vinska industrija

Kiselost vina obično se tehnički naziva titrabilna kiselost. Titrabilna kiselost je povezana je s pH vrijednošću, ali mjere različite stvari. Titrabilna kiselost mjeri količinu kiseline, dok pH mjeri jačinu kiselosti. Vino koje ima visok nivo kiseline će imati nisku razinu pH. Vina s visokom kiselinom su stabilnija jer imaju manje bakterija. Dvije glavne kiseline koje se nalaze u grožđu su vinska kiselina i jabučna kiselina. Vina s visokom kiselinom imaju jači okus na nepcu, dok vino s niskom kiselinom su nježnija i osjećaju se glađe na nepcu. PH vrijednost vina se kreće između 3 i 4. Važno je napomenuti da su brojevi na skali lakmus papira logaritamski, a ne linearni, odnosno vrijednost pH 5 nije dva puta kiseliji od vrijednosti pH 6 nego je čak 10 puta kiseliji.

Vina s visokim kiselinama se vremenom bolje poboljšavaju nego vina s manjim količinama kiseline. Dok vina s visokom razinom pH su sklonija zagađivanju. Mikrobi zamućuju vina koja imaju visoku pH razinu. Ovaj problem se rješava dodavanjem sumpor-dioksida, koji pomaže apsorbiranju djela oksidacije koja hrani mikrobe. Međutim  potrebno je mnogo više sumpornog dioksida kako bi se dobio isti efekt u vinu na razini pH 4 kao onom s 10 puta većom kiselošću nego na nivou pH 3.

Kiselost vina počinje u samom vinogradu. Kalij koji se nalazi u zemlji ulazi u grožđe i povećava lužnatost, što pomaže u neutralizaciji kiselosti i podiže pH. Grožđe koje nije zrelo ima viši nivo kiseline i taj nivo kiselosti opada kako grožđe sazrijeva. Kiselost vina se može povećavati dodavanjem vinske kiseline u sok od grožđa prije fermentacije. Ako vino sadrži puno vinske kiseline kada se flašira, ona se može ohladiti u kristale. Kristali, koji se ponekad nazivaju vinski dijamanti se rastvaraju kako se vino zagrijeva.

## 2.2. Tradicionalne metode mjerenja parametara vina

Titracija s natrijevim hidroksidom je metoda kojom se mjeri kiselost vina. Uzorak vina se pomiješa s indikatorom, a zatim se postepeno dodaje natrijev hidroksid dok se boja ne promijeni. Količina dodanog natrijevog hidroksida bilježi se i koristi za izračunavanje kiselosti vina.

Druga metoda, mnogo primitivnija, je kušanje, odnosno subjektivno procjenjivanje kiselosti vina putem degustacije. Vinski stručnjaci procjenjuju kiselost vina na temelju okusa. Visoka kiselost vina se može osjetiti sa žarenjem na jeziku i slinjenjem. Metoda naravno nije precizna kao ostale ali može pružiti općenitu sliku o kiselosti vina.

Te metoda pH papira: pH papir je jednostavan i jeftin način mjerenja kiselosti vina. Papir se umoči u uzorak vina, a boja koju papir promijeni ukazuje na pH vrijednost. Ova metoda može dati samo približnu vrijednost pH, ali može biti korisna za brze provjere kiselosti.

Postoje još razne metode za mjerenje kiselosti vina kao što su titracija s otopinom sumporne kiseline te modernije metode kao pH elektroda za vino koja u sebi ima mikročip i potrebne senzore za mjerenje kiselosti vina te još mnoge druge metode.

# 3. PRIMJENJENE TEHNOLOGIJE I KOMPONENTE

## 3.1. ESP32-CAM

*ESP32-CAM* (Slika ***3.1.*** ) je jako mali modul s kamerom koji ima ugrađeni *ESP32-S* čip. *ESP32* je serija mikrokontrolera male potrošnje na čipu s integriranom bežičnom mrežom (engl. *WiFi)* i *Bluetoothom*. Pored *OV2640* kamere i nekoliko ulaza/izlaza opće namijene (engl. *GPIO*) za povezivanje periferije, također ima ugrađen utor za mikro memorijsku karticu (engl. *microSD*) koja može služiti za spremanje podataka uslikanih pomoću kamere za daljnje moguće potrebe. Ovaj modul je korišten u završnom radu za slikanje lakmus papira te dobivanje pH vrijednosti vina. Jedan od glavnih razloga zašto je korišten ovaj modul su:

- Košta 10 eura, odnosno jeftin je
- Ima ugrađen *WiFi* i *Bluetooth*
- Jednostavan za koristiti
- Moguće pohranjivanje slika i podataka
- Omogućeno instaliranje i pokretanje *MicroPythona*


***Sl. 3.1.*** *ESP32-CAM* [6]

Spomenuti mikrokontroler sadrži više ulazno/izlaznih pribadača (engl. *pins*) koji su vidljivi na samom mikrokontroleru, gdje pokraj svakog pribadača piše njegova oznaka, te oznake olakšavaju u spajanju mikrokontrolera s ostalim perifernim uređajima. Postoje pribadači za napon (to su pribadači *3V3* gdje je napon 3.3 volta, 5 voltni pribadači (*VCC* pribadač)), također postoje pribadači za spajanje uzemljenja (engl. *GND*) mikrokontrolera te sami pribadači za komunikaciju s računalom i perifernim uređajima. Prema slici ***3.2.*** se vidi raspored pribadača *ESP32*.


***Sl. 3.2.*** *Raspored pribadača ESP32-S* [7]

Mikrokontroler *ESP32-CAM* se programira u razvojnom okruženju koje se zove *Arduino* razvojno okruženje koje je objašnjeno u nastavku.

## 3.2. Arduino razvojno okruženje

*Arduino* je softver otvorenog koda koji služi za pisanje koda i slanje istog na *Arduino* mikroupravljače. *Arduino* je nastao od talijanske tvrtke *SmartProjects* 2005. *Arduino* razvojno okruženje pruža jednostavno sučelje za pisanje, uređivanje i prenošenje programskog koda na *Arduino* ploču i također na *ESP32*. Pruža jednostavan i pristupačan način za programiranje mikrokontrolera i stvaranje interaktivnih uređaja. Nakon instalacije *Arduino* razvojnog okruženja potrebno je instaliranje *ESP32* datoteka kako se mogao koristiti mikrokontroler. Također u tom programskom okruženju postoje već gotovi kodovi koji su već napisani i služe za testiranje mikrokontrolera, te je ovo preporučena praksa raditi kako bi bilo ustanovljeno da svaka komponenta na *ESP32* ispravno radi.

## 3.3. Python

*Python* je korišten za obradu slike, odnosno za dobivanje parametara slike te određivanje rezultata mjerenja. *Python* je programski jezik opće namjene, interpretiran i visoke razine kojeg je napravio Guido van Rossum 1990. godine. *Python* dopušta programerima korištenje nekoliko stilova programiranja. Objektno orijentirano, strukturno i aspektno orijentirano programiranje stilovi su dopušteni korištenjem *Pythona* te ova fleksibilnost čini *Python* programski jezik sve popularnijim. *Python* se najviše koristi na *Linuxu*, no postoje i inačice za druge operacijske sustave. *Python* je popularan jezik u obradi slika zbog svoje široke podrške biblioteka, jednostavnosti koda, velike zajednice, integracije s drugim alatima i portabilnosti. Postoji vise razloga korištenja *Pythona* u obradi slika, a najčešći razlozi su:

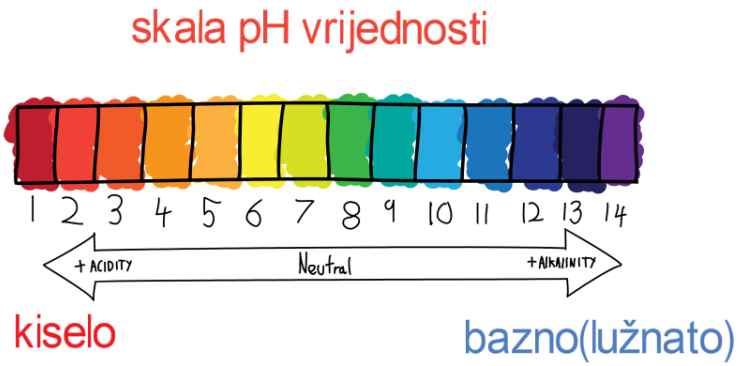
- Široka podrška biblioteka
- Jednostavnost koda
- Velika zajednica i dokumentacija
- Integracija s drugim alatima
- Portabilnost

## 3.4. Lakmus papir

Lakmus papir (Slika *3.3.* ) je komponenta preko koje se dobila pH vrijednost vina. Lakmus papir je jednostavan papirni indikator koji se koristi za brzo i jeftino testiranje pH vrijednosti otopina. To je jedan od najčešće korištenih indikatora u laboratorijskim, industrijskim i kućnim postavkama. Lakmus papir se dobiva od suhih i mljevenih lišajeva *Roccella tinctoria* ili *Roccella fuciformis*, koji sadrže pigmente osjetljive na kiseline i baze. Ti pigmenti se impregniraju u papir, stvarajući lakmus papir koji mijenja boju ovisno o pH vrijednosti tekućine koju testira. Lakmus nanesen na vrpcu filtarskog papira mnogo se koristio kao kiselo-lužnati indikator koji se u kiseloj tekućini oboji crveno, a u lužnatom u ljubičastu boju.

***Sl. 3.3.*** *Lakmus papir* [11]

Ph skala (Slika *3.4* )je numerička skala koja se koristi za mjerenje kiselosti ili alkalnosti otopina. PH skala ima raspon od 0 do 14, pri čemu 7 predstavlja neutralnu pH vrijednost. Lijeva strana skale (pH 0) označava jako kiselu tekućinu, dok desna strana skale označava lužnatu tekućinu (pH 14), odnosno boje idu od crvene (kiselo) sve do ljubičaste (lužnato).




***Sl. 3.4*** *pH skala* [12]

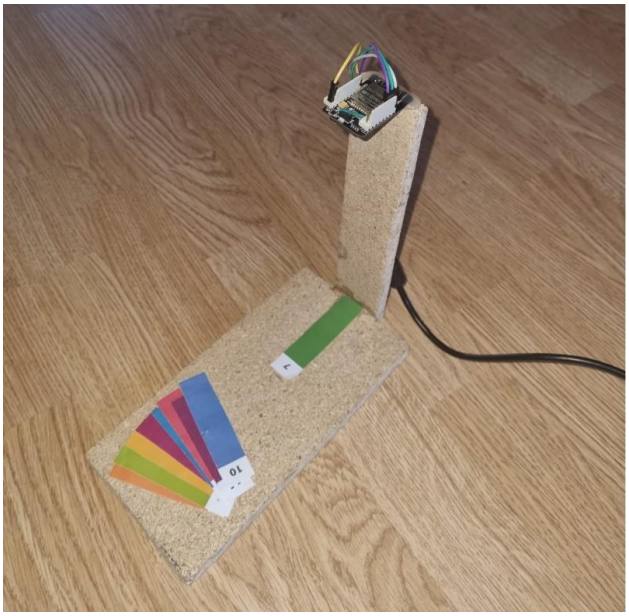
Lakmus papir je brz, jednostavan i jeftin način za procjenu pH vrijednosti otopina. Iako nije najpreciznija metoda, može biti vrlo korisna u mnogim situacijama, posebno za brze provjere pH vrijednosti u kućanstvu, laboratoriju ili industrijskim postavkama.

# 4. IZRADA SUSTAVA

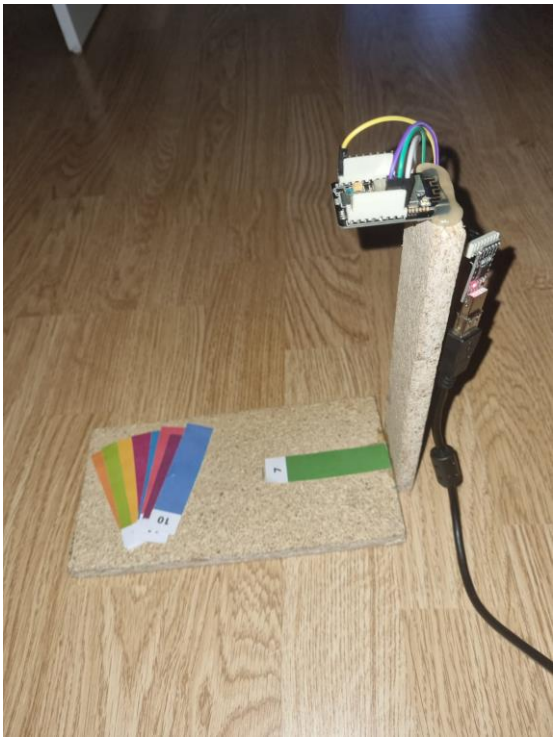
Za izvedbu uređaja za mjerenje pH vrijednosti vina bilo je potrebno izraditi maketu odnosno fizičku izvedbu uređaja, programiranje *ESP32-CAM* u *Arduino* razvojnom okruženju programskom okruženju te korištenje *Python* programskog jezika za obradu slike i za ispisivanje rezultata mjerenja.

## 4.1. Fizička izvedba

Praktični dio ovog završnog rada je izrada makete za mjerenje pH vrijednosti vina (Slika ***4.1.*** i Slika ***4.2.*** ). To je postignuto korištenjem šperploče koja je služila za izradu same makete. Maketa je sastavljena od dva dijela, prvi dio je ravna ploča na kojoj se vrši mjerenje pH vrijednosti, odnosno na tu ploču stavljamo lakmus papir (za testiranje je korišten papir u boji). Drugi dio je postolje za kameru koji je podignuti 20 cm od ravne ploče zbog dobivanja boljih rezultata mjerenja. Kada bi kamera bila previše blizu lakmus papira onda ne bi dobili dobre rezultate mjerenja zato što bi bilo malo vanjskog svijetla te zbog toga ne bi dobili dobre rezultate mjerenja. Pronađena je idealna udaljenost kamere od lakmus papira kako bi bilo prostora da vanjska svjetlost obasja lakmus papir. Nije korišten dodatni izvor svjetlosti jer se pokazalo kako prirodna svjetlost najbolje utiče na mjerenje. Umjetni izvor svjetlosti bi bio previše jak te bi se svjetlost odbijala od papir. Dva dijela šperploče su spojena s ljepilom i ekserima kako bi dodatno učvrstili sustav. Kamera odnosno *ESP32-CAM* je zalijepljen s vrućim lijepkom kako bi kamera bila stabilna tijekom mjerenja pH vrijednosti. Na samo ploči gdje se stavlja lakmus papir je označeno mjesto gdje se stavlja isti, odnosno gdje kamera pokriva područje koje obrađuje, to je jako važno jer je kod napisan tako da se rezultati mjerenja dobivaju na osnovi centra kamere te se lakmus papir mora postaviti u centar područja kojeg kamera obuhvaća. Cijeli sustav je spojen pomoću *USB* (univerzalna serijska sabirnica) kabela za računalo radi napajanja, također je moguće korištenje dodatnog izvora napajanja npr. baterije, jer sustav čitav radi uz pomoć *WiFi* mreže. Na sljedećim slikama vidimo kako je maketa izrađena te kako čitav sustav izgleda.



***Sl. 4.1.*** *Izgled makete*



***Sl. 4.2.*** *Izgled makete s boka*

## 4.2. Princip rada sustava

Problem je riješen s dva programska koda, prvi je s programskim jezikom C tako da je *ESP32-CAM* korišten za snimanje slika i posluživanje istih putem web servera *Prilog A.* , odnosno video prijenos s kamere na web server, te kasnije obrada tog video prijenosa s *Pythonom Prilog B.* .

### 4.2.1. Programski kod za upravljanje *ESP32-CAM*

Prvo je potrebno spojiti *ESP32-CAM* na računalo. To je postignuto korištenjem *FTDI* programatora koji pojednostavljuje konekciju *ESP*-a s računalom jer korišteni *ESP* nema ugrađeno *USB* sučelje za komunikaciju s računalom. Kada je uspješno spojen *ESP32-CAM* potrebno ga je programirati za rješavanje problema prijenosa videa na mrežni (eng. *web*) server. Prvo su uključene biblioteke koje omogućuju različite funkcije. Korištene biblioteke su:

- *WebServer.h* – biblioteka za implementaciju web servera na *ESP32*
- *Wifi.h* – biblioteka za upravljanje *WiFi* vezom
- *esp32cam.h* – biblioteka koja pruža podršku za *ESP32-CAM* modul

Nakon toga je potrebno konfigurirati *WiFi* mrežu gdje je definiran naziv mreže *(*eng. *SSID)* i lozinka za *WiFi* mrežu kojoj će se *ESP32-CAM* povezati. Te je nakon toga inicijaliziran mrežni server na portu 80 sto vidimo prema slici ***4.3***.

```
const char* WIFI_SSID = "********";
const char* WIFI_PASS = "********";

WebServer server(80);
```

***Sl. 4.3.*** *Konfiguracija mreže i inicijalizacija mrežnog servera*

Sljedeće je definiranje razlučivosti slike gdje se definiraju tri rezolucije prema slici ***4.4.*** .

```
static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
```

***Sl. 4.4.*** *Definiranje razlučivosti slike*

Zatim je napisana funkcija *serveJpg()* koja služi za snimanje slike pomoću *ESP32-CAM* modula. Ako snimanje slike ne uspije, server će vratiti odgovor s kodom 503. Ako je snimanje slike uspješno, slika se šalje klijentu kao odgovor s kodom 200, prema slici *4.5.* .

```
void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame->getHeight(),
                static_cast<int>(frame->size()));

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
  WiFiClient client = server.client();
  frame->writeTo(client);
}
```

***Sl. 4.3.*** *Funkcija serveJpg()*

Također su napisane funkcije za rukovanje zahtjevima slika određene rezolucije: *handleJpgLo()*, *handleJpgHi()*, *handleJpgMid()*. Ove funkcije postavljaju rezoluciju kamere na odgovarajuću rezoluciju (nisku, visoku ili srednju) i zatim pozivaju *serveJpg()* za posluživanje slike prema slici *4.6.*

```
void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi()
{
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid()
{
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}
```

***Sl. 4.4.*** *Funkcije za postavljanje rezolucije kamere*

Funkcija *setup()* postavlja serijski komunikacijski priključak za ispis poruka, konfigurira *ESP32-CAM* modul, postavlja rezoluciju kamere, broj buffera i kvalitetu *JPEG* kompresije. Konfigurira *Wi-Fi* vezu, čeka da se poveže na mrežu i ispisuje *IP* adresu na serijski priključak. Registrira putanje za zahtjeve slika određene rezolucije i pokreće web server prema slici *4.7*.

```
void  setup(){
  Serial.begin(115200);
  Serial.println();
  {
    using namespace esp32cam;
    Config cfg;
    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
  }
  WiFi.persistent(false);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Serial.print("http://");
  Serial.println(WiFi.localIP());
  Serial.println("  /cam-lo.jpg");
  Serial.println("  /cam-hi.jpg");
  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);
  server.on("/cam-hi.jpg", handleJpgHi);
  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();
}
```

***Sl. 4.5.*** *Funkcija setup()*

Na kraju funkcija *loop()* koja provjerava i obrađuje dolazne zahtjeve mrežnog servera.

Ovaj kod omogućuje pristup *ESP32-CAM* modulu putem web preglednika i omogućuje prikaz slika snimljenih kamerom u različitim rezolucijama (niska, srednja, visoka).

### 4.2.2. Programski kod za obradu slike

Napisan je *Python* kod koji koristi *OpenCV* biblioteku za obradu slika i boja te pristupa *IP* kameri. Slike se dohvaćaju s kamere putem *URL*-a i prikazuju na zaslonu. Također se određuje boja središnjeg piksela slike te se prikazuje pH vrijednost boje na slici.

Prvo je bilo potrebno uvesti biblioteke i definiranje polja za boje: U kodu su uvezene biblioteke *array*, *cv2*, *urllib.request*, *numpy* i *pandas*. Definiraju se polja *ra*, *ga* i *ba* koja sadrže vrijednosti *RGB* (model boja) komponenti boja prema slici *4.8.* .

```
import array as arr
import cv2
import urllib.request
import numpy as np
import pandas as pd

ra = arr.array('i', [238,238,245,251,244,163,77,1,5,81,69,42,148,123])
ga = arr.array('i', [55,52,126,169,236,205,184,146,148,117,74,47,36,39])
ba = arr.array('i', [34,121,38,35,8,57,71,71,149,186,159,132,140,121])
```

***Sl. 4.6.*** *Uključivanje bibilioteka i inicijalizacija polja*

Zatim je postavljen *URL* koji sadrži adresu za pristup slikama s *IP* kamere te je napravljen prozor za prikaz boja. Te je također kreiran *VideoCaputure* objekt za pristup video prijenosu s kamere prema slici *4.9*.

```
url = 'http://192.168.2.42/cam-hi.jpg'
cv2.namedWindow("Color recognition", cv2.WINDOW_AUTOSIZE)

# Kreiranje VideoCapture objekta
cap = cv2.VideoCapture(url)
```

***Sl. 4.7.*** *URL za kameru, prozor za prikaz boja i VideoCapture objekt*

Sljedeće je definirana funkcija *getColorPH()* za određivanje pH vrijednosti pojedine boje. Funkcija *getColorPH()* prima vrijednosti *R*, *G* i *B* komponenti boje i vraća indeks boje (pH) na temelju najmanje udaljenosti od referentnih boja prema slici *4.10*. Indeks boje označava pH vrijednost boje.

```
def getColorPH(R,G,B):
    indeks = 1
    minimum = 10000
    for i in range(14):
        d = (0.3*pow((R-ra[i]),2)) + (0.59*pow((G-ga[i]),2)) + (0.11*pow((B-ba[i]),2))
        if(d<=minimum):
            minimum = d
            indeks = i+1
    return indeks
```

***Sl. 4.8****. Funkcija getColorPH()*

Bilo je potrebno i provjeriti uspostavu prijenosa prema slici *4.11*. . Ako se ne uspije otvoriti video prijenos s kamerom, ispisuje se poruka i program završava.

```
# Provjeri da li je prijenos uspostavljen
if not cap.isOpened():
    print("Failed to open the IP camera stream")
    exit()
```

***Sl. 4.9.*** *Provjera da li je prijenos uspostavljen*

Te glavna petlja za ispisivanje slike u kojoj se čita jedan frame s video prijenosa. Koriste se funkcije *urllib.request.urlopen()* i *cv2.imdecode()* za dohvat slike s kamere i dekodiranje u oblik prikladan za obradu. Zatim se izračunao središnji piksel slike i dohvaćaju se njegove *RGB* I *HSV* (model boja) vrijednosti boje. Na temelju *RGB* vrijednosti boje, koristi se funkcija *getColorPH()* za određivanje pH vrijednosti. Na slici se iscrtavaju pravokutnik, tekst s pH vrijednošću i krug koji označava središte slike. Prikazuje se obrađeni frame u prozoru "*Frame*". Ako se pritisne tipka "*Esc*" petlja se prekida. Na samom kraju programa oslobađamo resurse (video prenos) i zatvaraju se prozori prema slici ***4.12***.

```
while True:
    # Čitaj jedan frame s video prenosa
    img_resp=urllib.request.urlopen(url)
    imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)

    im = cv2.imdecode(imgnp,-1)

    frame = im
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    height, width, _ = frame.shape

    cx = int(width / 2)
    cy = int(height / 2)

    pixel_center = hsv_frame[cy, cx]
    hue_value = pixel_center[0]

    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]), int(pixel_center_bgr[2])

    ph = getColorPH(r,g,b)

    cv2.rectangle(frame, (cx - 120, 50), (cx + 200, 120), (255, 255, 255), -1)
    cv2.putText(frame, "ph vrijednost: " + str(ph), (cx - 90, 100), 0, 1, (b, g, r), 3)
    cv2.circle(frame, (cx, cy), 5, (25, 25, 25), 3)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1)
    if key == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

***Sl. 4.10.*** *Glavna petlja*

# 5. TESTIRANJE SUSTAVA I PRIKAZ REZULTATA

Pri testiranju sustava korišten je papir u boji koji predstavlja lakmus papir. Papir u boji je korišten za testiranje radi pojednostavljivanja zadatka jer je lakše isprintati skalu pH vrijednosti s njihovim pripadajućim bojama nego korištenje lakmus papira. Kada bi bio korišten lakmus papir bilo bi potrebno (za testiranje) umočiti u 14 različitih tekućina (od jako kiselih pa sve do lužnatih tekućina) kako bi dobili rezultate mjerenja. To bi napravilo jako velik posao i pitanje je da li bi uopće bilo moguće dobiti odnosno kupiti različite tekućine s različitim pH vrijednostima, te je zbog toga korišten papir s bojama kao reprezentacija lakmus papira, gdje na svakom komadiću papira imamo isprintanu koja je pH vrijednost.

Prema slici ***5.1*** prikazani su rezultati mjerenja. Dobiveni rezultati su prilično dobri. Od 14 mogućih vrijednosti sustav je 11 vrijednosti izmjerio točno dok je 3 vrijednosti (pH 1, pH 8 i pH 12) interpretirao pogrešno. Pogreške se javljaju zbog vanjske svjetlosti. Naime, kako se vanjska svjetlost mijenja, mijenjaju se i rezultati mjerenja, jer je kamera jako osjetljiva na promjenu svjetline, odnosno, kada se smanji svjetlina koja obasjava lakmus papir (papir u boji) onda boja izgleda tamnija nego što je, te to utiče na funkciju koja određuje pH vrijednost jer se *r,g,b* vrijednosti mijenjaju promjenom svjetline. Pogreške ne odudaraju mnogo od stvarnih vrijednosti te sustav radi dobro, odnosno za pH vrijednost 1 sustav je izmjerio pH vrijednost 3, to je zato što su boje od tih pH vrijednosti jako slične te je bilo promjena pH vrijednosti kada se vanjska svjetlost mijenjala. Otprilike postotna uspješnost sustava je 78%, ta vrijednost se može mijenjati jer će nekad sustav izmjeriti točnih npr. 14 od 14 vrijednosti, a nekada će izmjeriti 8 točnih od 14 vrijednosti, sve zbog promjene svjetline.


***Sl. 5.1***. *Rezultati mjerenja*

# 6. ZAKLJUČAK

U ovom radu su objašnjene različite tehnologije za mjerenje parametara vina primjenom računalnog vida, izgrađen je sustav za mjerenje istog te su prikazani rezultati mjerenja za koje je dobivena određena točnost.

Na temelju rezultata dobivenih u ovom istraživanju mjerenja pH vrijednosti vina može se zaključiti da se u radu postigao rezultat od ukupno 78% uspješnosti u predviđanju pH vrijednosti vina. Iako rezultati nisu potpuno točni, to ukazuje na određenu razinu prediktivne sposobnosti metode koja je korištena. Postoji mnogo faktora koji utječu na rezultate, a glavni faktor je vanjska svjetlina koja mijenja boju odnosno *r,g,b* vrijednosti boje te se tako mijenja i pH vrijednost.

Unatoč postignutoj uspješnosti, rezultati istraživanja ukazuju na potrebu za daljnjim istraživanjem i poboljšanjem metode mjerenja pH vrijednosti vina pomoću računalnog vida. Moguće je istražiti nove tehnike mjerenja ili koristiti dodatne faktore za poboljšanje preciznosti tako da se eliminira problem s vanjskom svjetlosti i koristi modificirana umjetna svjetlost kako bi se dobili sto precizniji rezultati.

Iako trenutna uspješnost mjerenja nije idealna, tehnika mjerenja pH vrijednosti vina pomoću računalnog vida može pružiti općenite informacije u industriji vinarstva. To može biti korisno za praćenje kvalitete vina, identifikaciju abnormalnosti u proizvodnji ili optimizaciju procesa fermentacije.

# 7. LITERATURA


[1] »vino.ba,« 2021. [Mrežno]. Available: https://vino.ba/sta-je-kiselost-u-vinu/. [Pokušaj pristupa Srpanj 2023].

[2] »Wikipedia ESP32,« 2023. [Mrežno]. Available: https://en.wikipedia.org/wiki/ESP32. [Pokušaj pristupa Srpanj 2023].

[3] T. Mokrović, »HANNA instruments,« 2022. [Mrežno]. Available: https://blog.hannaservice.eu/hr/vaznost-mjerenja-ph-vrijednosti-vina/. [Pokušaj pristupa Srpanj 2023].

[4] »Random nerd tutorials,« [Mrežno]. Available: https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/. [Pokušaj pristupa Srpanj 2023].

[5] »Random nert tutorials,« [Mrežno]. Available: https://randomnerdtutorials.com/getting-started-with-esp32/. [Pokušaj pristupa Srpanj 2023].

[6] T. B.V., »TinyTronics,« 2014-2023. [Mrežno]. Available: https://www.tinytronics.nl/shop/en/development-boards/microcontroller-boards/with-wi-fi/esp32-cam-wifi-and-bluetooth-board-with-ov2640-camera. [Pokušaj pristupa Srpanj 2023].

[7] »PiShop.ca,« 2023. [Mrežno]. Available: https://www.pishop.ca/product/esp32-cam-camera-module-based-on-esp32/. [Pokušaj pristupa Srpanj 2023].

[8] »Wikipedia Arduino,« 2018. [Mrežno]. Available: https://hr.wikipedia.org/wiki/Arduino. [Pokušaj pristupa Srpanj 2023].

[9] »Wikipedia Python,« 2022. [Mrežno]. Available: https://hr.wikipedia.org/wiki/Python_(programski_jezik). [Pokušaj pristupa Srpanj 2023].

[10] »Wikipedia Lakmus,« 2021. [Mrežno]. Available: https://hr.wikipedia.org/wiki/Lakmus. [Pokušaj pristupa Srpanj 2023].

[11] D. Michaud, »911Metallurgist,« 2012 - 2023. [Mrežno]. Available: https://www.911metallurgist.com/blog/litmus-paper-for-ph-measurement. [Pokušaj pristupa Srpanj 2023].

[12] V. Oslon, »Quantum of Science - Društvo za promociju prirodnih nauka "Nauka i svijet",« 2017. [Mrežno]. Available: https://quantumofjk.blogspot.com/2017/04/sta-je-ph-vrijednost-i-kako-je-odrediti.html. [Pokušaj pristupa Srpanj 2023].

# SAŽETAK

**Naslov**: Mjerenje parametara vina primjenom računalnog vida.

U ovom završnom radu izgrađen je sustav za mjerenje pH vrijednosti vina. Sustav se baziran na *ESP32-CAM* mikrokontroleru s ugrađenom kamerom koji služi za prikupljanje podataka, odnosno snimanje lakmus papira te slanja slika (video prenos) na mrežni server, na obradi slika pomoću programskog jezika *Python* pomoću kojeg su dobivene pH vrijednosti. Tehnologije koje su korištene pri izradi sustava su *Arduino* razvojno okruženje za programiranje uređaja i *Python* razvojno okruženje (*Visual Studio Code*) za obradu slike. Postignuti su rezultati sa 78% uspješnosti.

**Ključne riječi**: *Arduino, ESP32-CAM*, obrada slike, pH, vino

## ABSTRACT

Title: Measurement of wine parameters using computer vision.

In this final thesis, a system for measuring the pH value of wine was developed. The system is based on the ESP32-CAM microcontroller with a built-in camera, which is used for data collection, specifically capturing litmus paper images and sending them (video transmission) to a web server. The images are then processed using the *Python* programming language to obtain pH values. The technologies used in the system development include the Arduino IDE for device programming and the *Python* editor (Visual Studio Code) for image processing. The achieved results show a success rate of 78%.



Keywords: Arduino, ESP32-CAM, Image Processing, pH, Wine

# ŽIVOTOPIS

Ilija Jazvić rođen je 17. lipnja 2001. u Vinkovcima. U Odžaku (Bosna i Hercegovina) završava osnovnu školu “Vladimira Nazora“, te 2016. godine upisuje srednju školu „Pere Zečevića“ smjer tehničar računarstva u istom gradu. Godine 2020. ostvaruje upis na Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, smjer računarstvo. Godine 2023. sudjeluje na dva sajma inovacija Ilok i Ivanić Grad te osvaja zlatnu i brončanu medalju.

_____

Potpis autora

## PRILOG A: Programski kod za ESP32-CAM

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "********";
const char* WIFI_PASS = "********";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);

void serveJpg()
{
  auto frame = esp32cam::capture();
  if (frame == nullptr) {
    Serial.println("CAPTURE FAIL");
    server.send(503, "", "");
    return;
  }
  Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(), frame-
>getHeight(),
                static_cast<int>(frame->size()));

  server.setContentLength(frame->size());
  server.send(200, "image/jpeg");
  WiFiClient client = server.client();
  frame->writeTo(client);
}

void handleJpgLo()
{
  if (!esp32cam::Camera.changeResolution(loRes)) {
    Serial.println("SET-LO-RES FAIL");
  }
  serveJpg();
}

void handleJpgHi()
{
  if (!esp32cam::Camera.changeResolution(hiRes)) {
    Serial.println("SET-HI-RES FAIL");
  }
  serveJpg();
}

void handleJpgMid()
{
  if (!esp32cam::Camera.changeResolution(midRes)) {
    Serial.println("SET-MID-RES FAIL");
  }
  serveJpg();
}

void  setup(){
  Serial.begin(115200);
  Serial.println();
```

```
  {
    using namespace esp32cam;
    Config cfg;
    cfg.setPins(pins::AiThinker);
    cfg.setResolution(hiRes);
    cfg.setBufferCount(2);
    cfg.setJpeg(80);

    bool ok = Camera.begin(cfg);
    Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
  }
  WiFi.persistent(false);
  WiFi.mode(WIFI_STA);
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Serial.print("http://");
  Serial.println(WiFi.localIP());
  Serial.println("  /cam-lo.jpg");
  Serial.println("  /cam-hi.jpg");
  Serial.println("  /cam-mid.jpg");

  server.on("/cam-lo.jpg", handleJpgLo);
  server.on("/cam-hi.jpg", handleJpgHi);
  server.on("/cam-mid.jpg", handleJpgMid);

  server.begin();
}

void loop()
{
  server.handleClient();
}
```

*Prilog A.  programski kod za kameru*

## PRILOG B: Programski kod za obradu slike

```
import array as arr
import cv2
import urllib.request
import numpy as np
import pandas as pd

ra = arr.array('i', [238,238,245,251,244,163,77,1,5,81,69,42,148,123])
ga = arr.array('i', [55,52,126,169,236,205,184,146,148,117,74,47,36,39])
ba = arr.array('i', [34,121,38,35,8,57,71,71,149,186,159,132,140,121])

url = 'http://192.168.2.42/cam-hi.jpg'
cv2.namedWindow("Color recognition", cv2.WINDOW_AUTOSIZE)

# Kreiranje VideoCapture objekta
cap = cv2.VideoCapture(url)

def getColorPH(R,G,B):
    indeks = 1
    minimum = 10000
    for i in range(14):
        d = (0.3*pow((R-ra[i]),2)) + (0.59*pow((G-ga[i]),2)) + (0.11*pow((B-
ba[i]),2))
        if(d<=minimum):
            minimum = d
            indeks = i+1
    return indeks

# Provjeri da li je prijenos uspostavljen
if not cap.isOpened():
    print("Failed to open the IP camera stream")
    exit()

while True:
    # Čitaj jedan frame s video prenosa
    img_resp=urllib.request.urlopen(url)
    imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)

    im = cv2.imdecode(imgnp,-1)

    frame = im
    hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    height, width, _ = frame.shape

    cx = int(width / 2)
    cy = int(height / 2)

    pixel_center = hsv_frame[cy, cx]
    hue_value = pixel_center[0]

    pixel_center_bgr = frame[cy, cx]
    b, g, r = int(pixel_center_bgr[0]), int(pixel_center_bgr[1]),
int(pixel_center_bgr[2])

    ph = getColorPH(r,g,b)

    cv2.rectangle(frame, (cx - 120, 50), (cx + 200, 120), (255, 255, 255), -
1)
```

```
    cv2.putText(frame, "ph vrijednost: " + str(ph), (cx - 90, 100), 0, 1, (b,
g, r), 3)
    cv2.circle(frame, (cx, cy), 5, (25, 25, 25), 3)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1)
    if key == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

*Prilog B. programski kod za obradu slike*