DETEKCIJA POŽARA UPORABOM DIGITALNE OBRADE SIGNALA

Relja, Frane

Master's thesis / Specijalistički diplomski stručni

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:963059

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Repository of University Department of Professional Studies

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

Frane Relja

Z A V R Š N I R A D

Detekcija požara uporabom digitalne obrade slika

Split, lipanj 2020.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

Predmet: Digitalna obrada vizualnih medija

Z A V R Š N I R AD

Kandidat: Frane Relja

Naslov rada: Detekcija požara uporabom digitalne obrade slika

Mentor: dr.sc. Barbara Džaja

Split, lipanj 2020.

SADRŽAJ

Sažetak **1**

1. UVOD **2**

2. DIGITALNA OBRADA VIZUALNIH MEDIJA **3**

3. SUSTAV ZA DETEKCIJU POŽARA **5**

4. KOMPONENTE SUSTAVA **6**

4.1. Arduino Uno 7

4.2. BMP180 temperaturni senzor 10

4.3. Kamera 12

5. SHEMA SUSTAVA **13**

6. KREIRANJE HAAR-CASCADE KLASIFIKATORA **14**

6.1. Računalni vid 14

6.2. Haar-Cascade klasifikator 16

6.3. Kreiranje Haar-Cascade klasifikatora 16

7. KREIRANJE KORISNIČKE APLIKACIJE U VISUAL BASIC-u **24**

7.1. Visual Basic 2010 express edition 24

7.2. Emgu CV 26

7.3. Kreiranje forme 27

7.4. Programski kod 31

8. ARDUINO **37**

8.1. Arduino IDE 37

8.2. Programski kod 38

9. RAD SUSTAVA U STVARNIM UVJETIMA **40**

10. ZAKLJUČAK **44**

LITERATURA **45**

Sažetak

Detekcija požara uporabom digitalne obrade slika

U ovom radu opisan je sustav za detekciju požara uporabom digitalne obrade vizualnog medija, odnosno video signala s kamere u realnom vremenu korištenjem Microsoft Visual Basic-a. Kako bi osigurali sustav od lažnog uzbunjivanja, dodali smo temperaturni senzor (BMP180) povezan na Arduino mikrokontroler koji detektira porast temperature u prostoru, što je jedan od produkata gorenja. Nakon što su oba uvjeta zadovoljena (otkriven požarni objekt na video signalu s kamere i detektiran porast temperature iznad granične vrijednosti) korisnik je alarmiran e-mail porukom na željenu adresu. Sadržaj poruke je iznos temperature prostora u trenutku detekcije požara. Sustav je ispunio sva očekivanja te je pogodan za uporabu u kućanstvu i manjim prostorima.

Ključne riječ: Arduino, BMP180, detekcija, vatra, VB.Net

Summary

Fire detection using digital image processing

This paper describes a fire detection system using digital processing of visual media, respectively video signals from the camera in real time using Microsoft Visual Basic. To protect the system from false alarms, we added a temperature sensor (BMP180) connected to an Arduino microcontroller that detects a rise in room temperature, which is one of the products of combustion. After both conditions are met (a fire object is detected on the camera video signal and a rise of temperature above the limit value is detected) the user is alerted by e-mail to the desired address. The content of the message is the value of room temperature at the time of fire detection. The system met all expectations and is suitable for use in households and smaller spaces.

Keywords: Arduino, BMP180, detection, fire, VB.Net

1. UVOD

Vatra je otkrivena tisućama godina unatrag, ali ipak u današnjem svijetu gdje razvoj tehnologije i građevinarstva doživljava strmi rast još uvijek požari predstavljaju prijetnju ljudima i materijalnim dobrima. Požar je svako nekontrolirano gorenje čijim širenjem može doći do ugroze ljudi, životinja i imovine. Podaci CTIF-a (*engl. International Association of Fire and Rescue)* iz 2017. godine pokazuju da u svijetu godišnje izbije 3 503 425 požara u kojima život izgube 18 454 osobe što nisu zanemarive brojke [1]. Stoga, bez obzira koliko su današnje metode gradnje objekata sigurne u smislu zaštite od požara, uvijek postoji mogućnost za njegovo izbijanje te je neophodno osigurati njegovu ranu detekciju. Da bi to bili u stanju realizirati potrebno je poznavati način ponašanja i širenja požara u objektima.

Razvoj požara možemo podijeliti u četiri faze:

- početna faza,
- faza razvoja,
- razbuktala faza
- i faza gašenja,

gdje brzina širenja požara uvelike ovisi o vrsti objekta, uvjetima gorenja i količini požarnog opterećenja. Svaka slijedeća faza do faze gašenja podrazumijeva povećanje temperature i intenziteta požara, odnosno veću opasnost za ljude i strukturni integritet samog objekta te veću materijalnu štetu i troškove gašenja, stoga je naglasak na njegovom ranom otkrivanju.

Produkti gorenja su svjetlost, toplina, tlak i dim. Načini detektiranja požara su upravo prema ovim produktima. Veliki objekti javne i industrijske namjene koriste *Sprinkler* sustave za detekciju i gašenje požara koja predstavljaju značajna novčana ulaganja i iziskuju zapošljavanje stručnog osoblja za održavanje istih, dok u domovima možemo naići na razne dojavljivače požara koji detektiraju vatru porastom temperature u prostoru ili nastankom dima. Drugi način detekcije je uporaba termalnih kamera, ali kao što je već navedeno i za prošle sustave, njihova mana je izrazito visoka cijena. U ovom radu obradit će se otkrivanje požara prema detekciji intenziteta svjetlosti i porasta temperature te je ideja malim ulaganjem postići jednostavnu, brzu i pouzdanu detekciju požara korištenjem obične web kamere. Da bi sustav učinili efikasnijim i „pametnijim“ dodati će se i e-mail uzbunjivanje, a kao zaštita od lažnog dojavljivanja u sustav će se implementirati i temperaturni senzor koji detektira porast temperature u prostoriji.

2. DIGITALNA OBRADA VIZUALNIH MEDIJA

Digitalni vizualni mediji, kao što su slika ili video, su mediji koji su kodirani u strojno čitljivim formatima. Oni se mogu stvarati, gledati, distribuirati, modificirati i sačuvati na digitalnim elektroničkim uređajima. Digitalna obrada slike sastoji se od manipulacije slikama pomoću digitalnih računala. Njegova se upotreba eksponencijalno povećava u posljednjim desetljećima. Primjena ovog procesa kreće se od medicine pa sve do zabave, geološke obrade i daljinskog istraživanja. Multimedijski sustavi, jedni od stupova suvremenog informacijskog društva, u velikoj mjeri se oslanjaju na digitalnu obradu slika.

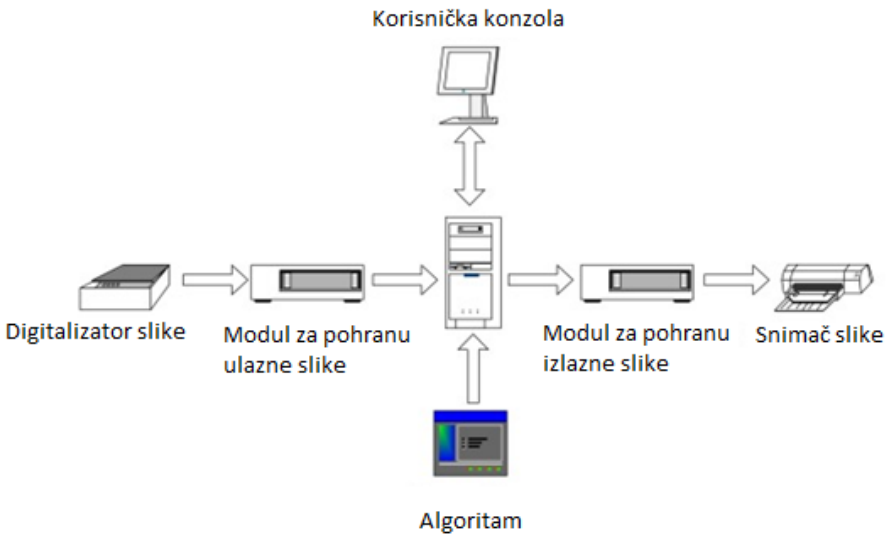
Disciplina digitalne obrade slike iznimno je rasprostranjena, uključuje tehnike digitalne obrade signala kao i tehnike specifične samo za slike. Slika se može smatrati funkcijom f (x, y) dviju kontinuiranih varijabli x i y. Da bi se takva funkcija digitalno obradila, treba ju uzorkovati i pretvoriti u matricu brojeva, odnosno potrebno je pretvoriti fizičku sliku u odgovarajuću digitalnu sliku te izlučiti značajne informacije iz digitalizirane slike primjenom različitih algoritama. Digitalna obrada slike uglavnom uključuje prikupljanje slika, obradu slike i analizu slike. Na svojoj osnovnoj razini, sustav za obradu digitalnih slika sastoji se od tri komponente:

- računalnog sustava na kojem se obrađuju slike,
- digitalizatora slike
- i uređaja za prikaz slike,

kao što je i prikazano na slici 2.1.

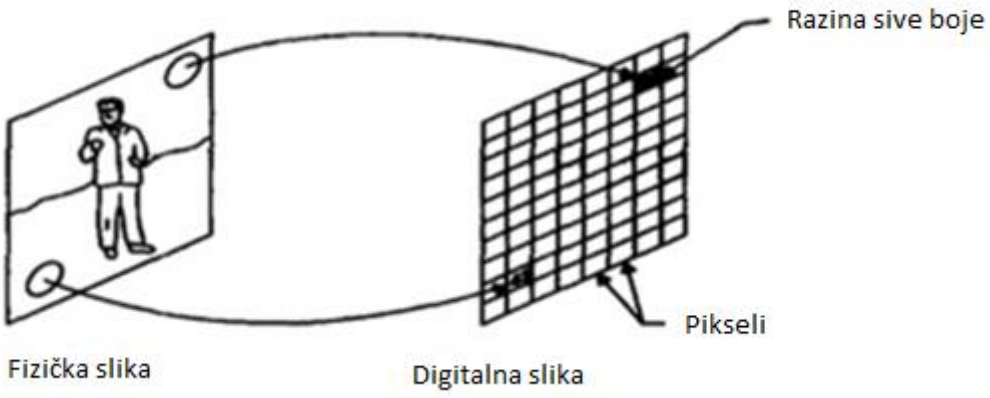
Budući da računalo predstavlja brojeve pomoću konačne preciznosti, te brojeve treba kvantizirati da bi se predstavili digitalno. Digitalna obrada slike sastoji se od manipulacije tim brojevima s ograničenom preciznošću. Obrada digitalnih slika može se podijeliti u nekoliko klasa: poboljšanje slike, obnavljanje slike, analiza slike i kompresija slike. Kod poboljšanja slike, slikom se manipulira tako da korisnik (osoba a ne računalo) može iz nje izvući korisne informacije. Tehnike obnove slika imaju za cilj obradu oštećenih slika iz kojih postoji statistički ili matematički opis razgradnje da bi se one mogle preokrenuti. Tehnike analize slike omogućuju obradu slike tako da se podaci automatski mogu iz nje izvući. Primjeri za analizu slike su segmentacija slike, ekstrakcija ruba te analiza teksture i pokreta. Važna karakteristika slika je ogromna količina informacija potrebnih za njihovo predstavljanje. Stoga, da bi bilo praktično pohranjivati i prenositi digitalne slike, potrebno je izvršiti neku vrstu kompresije, pri

čemu se redundantne informacije u slici iskorištavaju za smanjenje broja bita potrebnih za njihovo predstavljanje [2].



Slika 2.1 Sustav digitalne obrade slika

Na slici 2.2 dan je prikaz fizičke slike i pripadajuće digitalizirane slike. Fizičke slike podijeljene su u mala područja koja se nazivaju pikselima. Najčešće korištena podjela je metoda pravokutne mreže uzorkovanja u kojoj je slika segmentirana na više vodoravnih linija sastavljenih od susjednih piksela, a vrijednost svakog položaja piksela odražava svjetlinu odgovarajuće točke na fizičkoj slici [3].



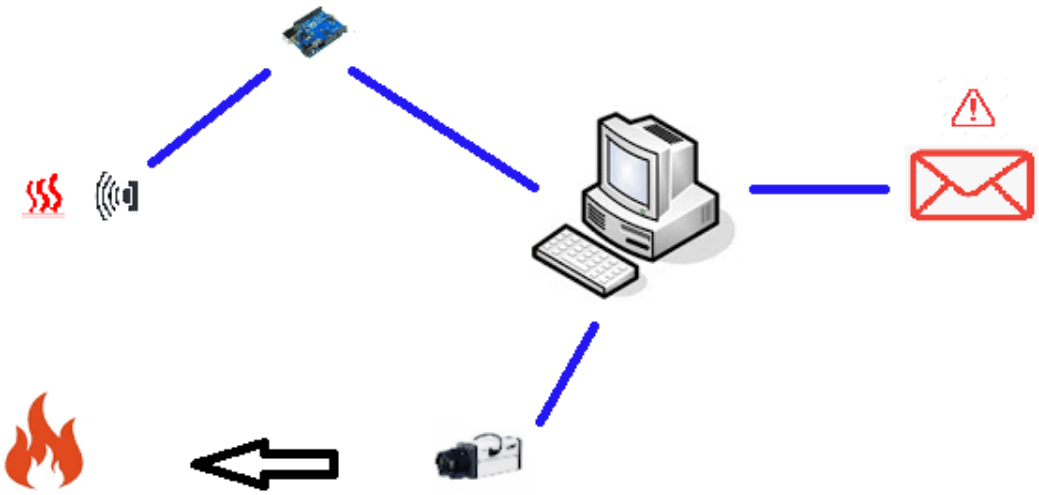
Slika 2.2 Prikaz fizičke i pripadajuće digitalizirane slike

3. SUSTAV ZA DETEKCIJU POŽARA

Glavna odlika ovakvog sustava bi bila pravovremeno i točno opažanje požara u prostoriji gdje je postavljeno. Kao što je već i spomenuto, kod požara, a poglavito u požarima zatvorenog prostora, ključnu ulogu čini vrijeme. Brza reakcija može osigurati smanjenje štete i spašavanje ljudskih života, stoga je potrebno prepoznati požar u njegovom začetku, što je ovaj sustav u mogućnosti. Program razvijen u Visual Basic-u u stvarnom vremenu obrađuje video signal sa web kamere te označava područje na kojem je opažena vatra. Pošto se kod ovakvog načina obrade vizualnog medija često zna pojaviti lažno očitanje potrebno je ovakav sustav zaštititi od istoga, jer inače sustav ne bi bio učinkovit.

Zaštita od lažne dojave osigurana je korištenjem temperaturnog senzora BMP180 koji, povezan na Arduino sučelje, očitava porast temperature u prostoriji. Poznato je da s pojavom vatre raste i temperatura okolnog prostora. Svaka prostorija u objektu ima svoj stupanj zagrijanosti koji može varirati ovisno o njenoj namjeni i vremenskim prilikama. Kada se odredi normalni raspon temperature za određenu prostoriju tada možemo postaviti temperaturnu granicu na kojoj će senzor, odnosno program, detektirati požar.

Bitno je naglasiti da kod ovog sustava treba voditi računa o mjestu postavljanja kamere i temperaturnog senzora. Kameru treba orijentirati prema prostoru kojeg želimo nadzirati (prostor koji je pod povećanom požarnom opasnosti) ali na dovoljnoj udaljenosti tako da, ukoliko bi došlo do požara, kamera ne bi bila vrlo brzo uništena, dok je temperaturni senzor potrebno postaviti što bliže štićenom prostoru tako da bi u što kraćem roku mogao osjetiti temperaturno povećanje. Ukoliko je potrebno, može se postaviti više senzora za veću osjetljivost, odnosno veći broj kamera za pokrivanje većeg prostora.



Slika 3.1 Model sustava za detekciju požara

Nakon što su oba uvjeta zadovoljena (slika 3.1: detektiran je plamen na slici te je temperaturni senzor očitao temperaturu veću od granične vrijednosti) korisnik se uzbunjuje putem e-mail poruke koja sadrži vrijednost očitane temperature sa senzora u trenutku detekcije požara.

4. KOMPONENTE SUSTAVA

Sustav za detekciju požara opisan u prethodnom poglavlju sastoji se samo od četiri komponente koje ćemo detaljno opisati u nastavku:

- Arduino UNO,
- BMP180 temperaturni senzor,
- web kamera,
- računalo,

što ga čini povoljnim i pogodnim za lako postavljanje u bilo kojem prostoru.

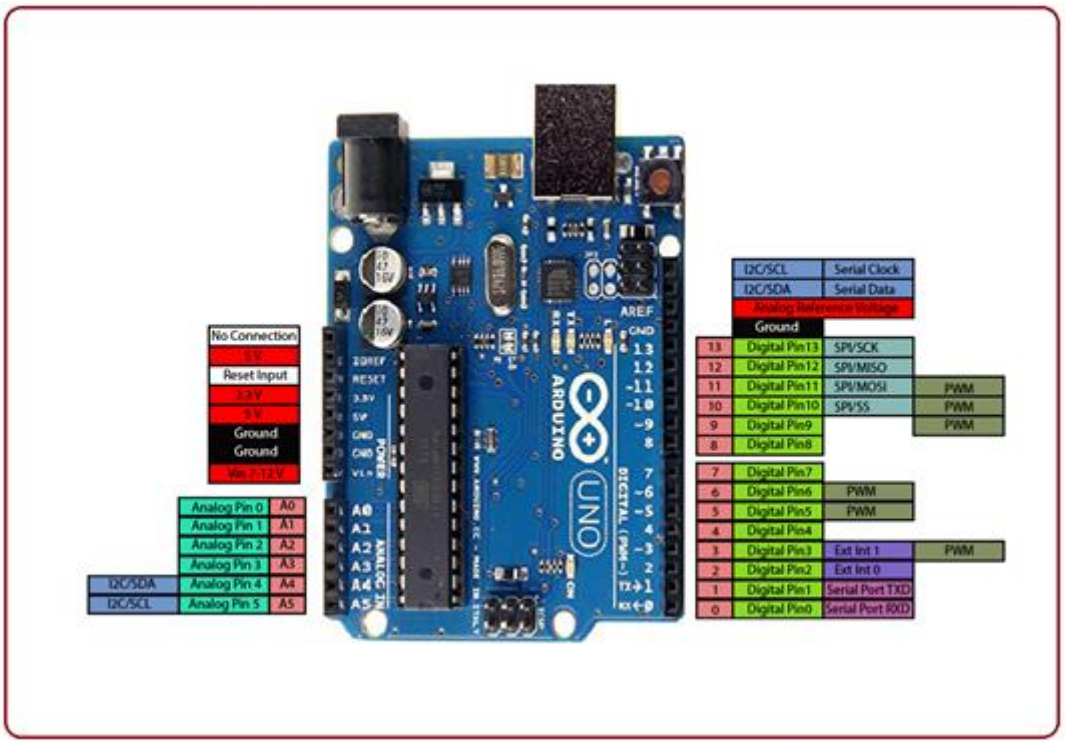
4.1 Arduino UNO

Arduino Uno je mikrokontroler otvorenog koda razvijen od strane *Arduino.cc*.-a koji se temelji na mikrokontroleru *Microchip ATmega328P*. Pločica je opremljena setovima digitalnih i analognih ulaza/izlaza (I/O) pinova koji se mogu povezati s različitim ekspanzijskim pločicama (štitnicima) i drugim integriranim krugovima. Arduino UNO ima 14 digitalnih I/O pinova (šest podržava PWM izlaz), 6 analognih I/O pinova i programira se pomoću Arduino IDE (integriranog razvojnog okruženja), putem USB kabela tipa B. Može se napajati USB kabelom ili vanjskom 9-voltnom baterijom, mada prihvaća napone između 7 V i 20 V. Slično je i s Arduino Nano i Leonardo.

Slika 4.1. Arduino UNO mikrokontroler [4]

Riječ "uno" znači "jedan" na talijanskom jeziku i odabrana je za označavanje početnog izdanja Arduino softvera. Pločica Uno prva je u nizu USB-ovih Arduino pločica; ona i inačica 1.0 Arduino IDE-a bile su referentne verzije Arduina, koje su se sada razvile u novija izdanja. ATmega328 na ploči dolazi unaprijed programiran s programom koji omogućava prijenos novog koda bez upotrebe vanjskog hardverskog programera. ATmega328 je mikrokontroler s jednim čipom kreiran od strane Atmel-a. Sadrži modificiranu 8-bitnu jezgru RISC procesora Harvard arhitekture. ATmega328kombinira 32 KB ISP flash memoriju s mogućnostima čitanja i pisanja, 1 KB EEPROM, 2 KB SRAM, 23 I/O linije opće namjene, 32 radna registra opće namjene, tri fleksibilna *timera* / brojila s usporednim modusima, unutarnjim i vanjskim prekidima, serijski programirajući USART, serijski ulaz SPI, 6-kanalni 10-bitni A/D pretvarač

(8-kanalni u TQFP i QFN/MLF paketima), programabilni zaštitni mjerač vremena s unutarnjim oscilatorom i pet načina softverskog odabira uštede energije. Uređaj radi na 1,8-5,5 volti te postiže propusnost približavajući se 1 MIPS po MHz.



Slika 4.2. Arduino UNO – prikaz pinova [5]

Arduino UNO sadrži slijedeće tehničke specifikacije:

- radni napon: 5 V,
- ulazni napon: 7 do 20 Volti,
- digitalni I/O pinovi: 14 (od kojih 6 može pružiti PWM izlaz) ,
- UART: 1,
- I2C: 1,
- SPPI: 1,
- analogni ulazni pinovi: 6,
- DC struja po I O pinu: 20 mA,
- DC struja za 3,3 V pin: 50 mA,
- flash memorija: 32 KB od čega 0,5 KB koristi bootloader,
- SRAM: 2 KB,

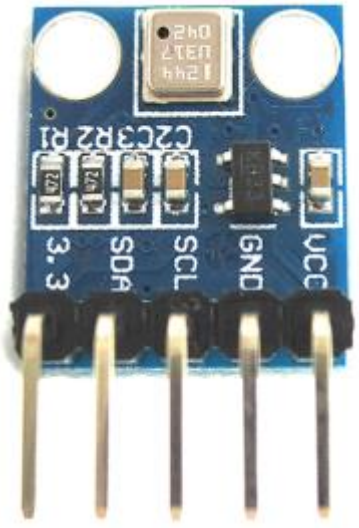
- EEPROM: 1 KB,
- brzina takta: 16 MHz,
- duljina: 68,6 mm,
- širina: 53,4 mm,
- težina: 25 g.

Generalne funkcije pinova:

- LED: Postoji ugrađeni LED koji pokreće digitalni pin 13. Kad je na pinu visoka vrijednost, LED svijetli, kad je na pinu niska vrijednost, on je isključen.
- VIN: ulazni napon na ploču Arduino/Genuino kada se koristi vanjski izvor napajanja (za razliku od 5 volti iz USB veze ili drugog reguliranog izvora napajanja). Napon se može opskrbiti kroz ovaj pin ili, ako se dovodi napon preko priključka za napajanje, pristupa mu se kroz ovaj pin.
- 5V: Ovaj pin izvodi reguliranih 5V od regulatora na ploči. Pločica se može napajati bilo iz istosmjerne utičnice (7 - 20V), USB priključka (5V) ili VIN pina na ploči (7-20V). Napajanje napona preko pinova od 5 V ili 3.3 V zaobilazi regulator, te može oštetiti ploču.
- 3V3: Napajanje od 3,3 volta koje generira ugrađeni regulator. Maksimalno strujno povlačenje je 50 mA.
- GND: pin za masu (uzemljenje).
- IOREF: Ovaj pin na ploči Arduino/Genuino pruža referentnu vrijednost napona s kojom mikrokontroler radi. Ispravno konfigurirani štit može očitati IOREF-ov napon i odabrati odgovarajući izvor napajanja ili omogućiti prevoditeljima napona na izlazima da rade s 5V ili 3.3V.
- Reset: dovođenjem ovog pina u stanje „LOW“ resetira se mikrokontroler [4].

4.2 BMP180 temperaturni senzor

BMP180 je jedan od senzora BMP XXX serije. Svi su dizajnirani za mjerenje barometrijskog tlaka ili atmosferskog tlaka. BMP180 je senzor visoke preciznosti dizajniran za široku primjenu u elektronici. Barometarski tlak nije ništa drugo nego težina zraka koja pritišće podlogu. Zrak ima težinu i gdje god ima zraka njegov se pritisak osjeća. Senzor BMP180 osjeti taj pritisak i pruža tu informaciju u digitalnom obliku. Također temperatura utječe na tlak, pa nam je potrebno očitanje tlaka s kompenzacijom temperature. Da bi se kompenzirao, BM180 ima i dobar temperaturni senzor te upravo zbog tog svojstva pogodan je za korištenje u ovom projektu, odnosno za potrebe rada iskoristit ćemo samo njegovu sposobnost mjerenja temperature dok ćemo mjerenja atmosferskog tlaka i nadmorske visine zanemariti.



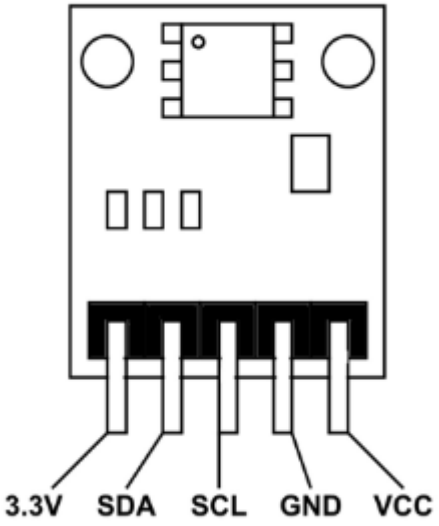
Slika 4.3 BMP180 temperaturni senzor [6]

Specifikacije BMP180 senzora su:

- Radni napon: 1.3V - 3.6V
- Ulazni napon: 3.3V do 5.5V
- Vršna struja: 1000uA
- Potrošnja u mirovanju: 0,1 uA
- Maksimalni napon kod SDA, SCL: VCC + 0,3V
- Radna temperatura: -40 ° C do + 80 ° C

Mnogo je razloga zbog kojih je BMP180 preferiran u odnosu na druge module kao što je izrazita preciznost pri mjerenju atmosferskog tlaka, mala potrošnja što omogućuje upotrebu ovog

senzorima u sustavima koji rade na baterijskom napajanju i brza komunikacija (podržava visoke brzine prijenosa podataka) [6].



Slika 4.4 BMP180 prikaz pinova [6]

Funkcija pinova prikazanih na slici 4.4:

- VCC: napajanje + 5V
- GND: masa (uzemljenje)
- SDA: serijski podatkovni pin (I2C sučelje)
- SCL: pin serijskog sata (I2C sučelje)
- 3.3V: ukoliko napajanje od + 5V nije dostupno modul se može napajati s ovog pina u iznosu od + 3,3 V

Kao što smo spomenuli, BMP180 senzor može mjeriti temperature od -40 °C do + 80 °C s preciznošću od ±2°C što je sasvim prihvatljivo u našem slučaju. Treba naglasiti ukoliko očekujemo veće temperature u prostoriji gdje postavljamo sustav za detekciju požara potrebno je upotrijebiti drugi senzor koji podnosi veće temperaturne oscilacije kao što je K-tip termopar i postaviti temperaturnu granicu sustava u skladu s tim uvjetima.

4.3 Kamera

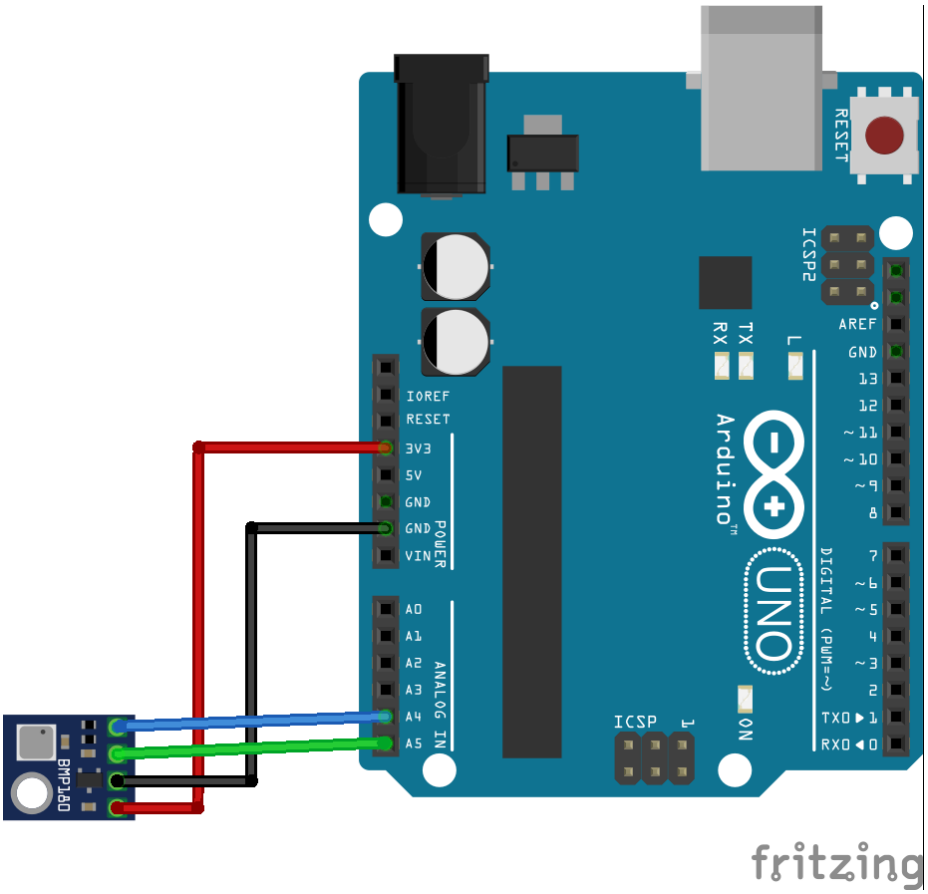
Za razliku od skupocjenih termalnih kamera koje se koriste u sustavima za detekciju požara ili kamera sa visokom rezolucijom, za ovaj projekt poslužiti će obična web kamera, lako dostupna i povoljna bez velikih zahtjeva. Model kamere korištene u ovom sustavu za detekciju požara je Genius Eye 110 web kamera sa slijedećim karakteristikama:

- Način povezivanja: žično
- Digitalni video format: WMV
- Max. digitalna video rezolucija: 640 x 480, 15 fps
- Sučelja: USB 2.0 - 4-polni USB Tip A
- Senzor slike: CMOS, 0.1 MP
- Podešavanje fokusa: ručno
- Format snimanja slike: JPEG
- Rezolucija fotografija: 160 x 120, 176 x 144, 320 x 240, 352 x 288, 640 x 480 [8]

Slika 4.6 Genius Eye 110 web kamera [8]

5. SHEMA SUSTAVA

Kao što je već spomenuto u prethodnim poglavljima ovaj sustav sa hardverske strane nema značajnih zahtjeva. Na slici 3.1 prikazan je model sustava koji predočuje njegovu jednostavnost stoga će u ovom poglavlju biti samo detaljnije prikazana shema spajanja temperaturnog senzora s Arduino UNO mikrokontrolerom. Shema spajanja je dana na slici 4.7.



Slika 4.7 Shema spajanja BMP180 modula i Arduino UNO mikrokontrolera [7]

Kao što je vidljivo na slici, shema spajanja je vrlo jednostavana. BMP180 se spaja na Arduino UNO na način da Vcc pin senzora povezujemo s Arduino-vim 5-voltnim pinom. Uzemljenje senzora povezujemo s Arduino-vom masom dok SDA pin temperaturnog senzora spajamo s analognim pinom broj A4 Arduino UNO-a. SCL pin BMP180 senzora spajamo s Arduino-vim analognim pinom broj A5. Da bi sustav bio potpun, potrebno je još povezati Arduino UNO mikrokontroler s računalom koristeći USB kabel te na isti način i web kameru. Za potrebe računala bitno je osigurati stalni izvor napajanja i internetsku vezu.

6. KREIRANJE HAAR-CASCADE KLASIFIKATORA

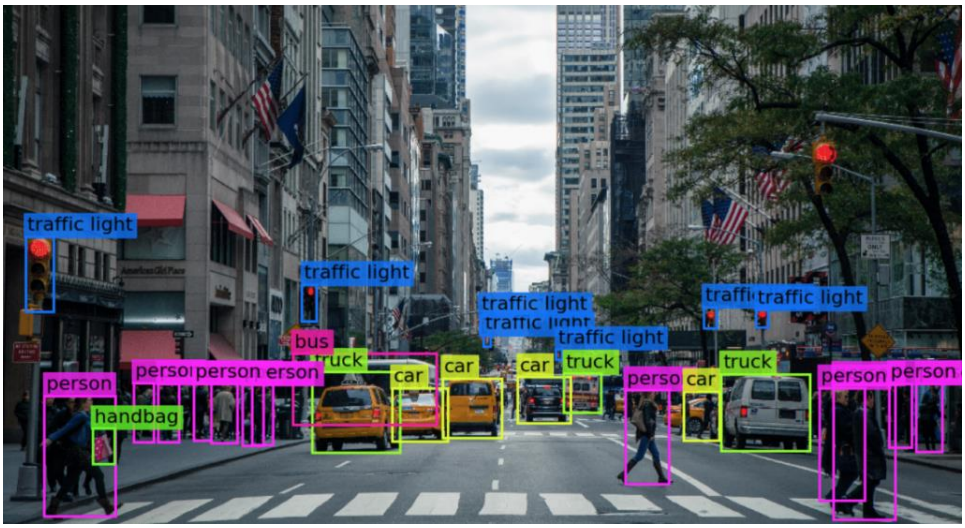
U ovom poglavlju biti će objašnjeno što je računalni vid, Haar-Cascade klasifikator i način stvaranja istog. Haar-Cascade klasifikator koristi se u Visual Basic aplikaciji za prepoznavanje vatre na slici. Drugim riječima, koristeći Haar-Cascade klasifikator „istreniramo“ program kako da prepozna požar na slici, odnosno u video zapisu.

6.1 Računalni vid

Prvo je potrebno pojasniti što je to računalni vid (*engl. Computer vision*) i neke od njegovih primjena.

Algoritmi računalnog vida trenutno su jedni od najsnažnijih AI sustava na svijetu. Sustavi računalnog vida nalaze uporabu u autonomnim vozilima, navigaciji robota, sustavima za prepoznavanje lica i još mnogo toga. Sustavi računalnog vida djeluju vrlo slično ljudskom vizualnom sustavu, tako što prvo razabiru rubove objekta, a zatim spajaju te rubove zajedno s objektom. Velika je razlika u tome što računalo tumači slike kao brojeve te je sustavu računalnog vida potreban način tumačenja pojedinih piksela koji sačinjavaju sliku.

Sustav računalnog vida dodijelit će vrijednosti pikselima na slici te ispitivanjem gradijenta vrijednosti između jedne regije piksela i druge regije piksela računalo može razabrati rubove. Na primjer, ako je određena slika sive boje, tada će se vrijednosti kretati od crne (predstavljene s 0) do bijele (predstavljene sa 255). Nagla promjena vrijednosti susjednih piksela pokazat će rub. Ovaj se osnovni princip uspoređivanja vrijednosti piksela može izvesti i s obojenim slikama. Sustav računalnog vida proučava vrijednosti piksela kako bi interpretirao sliku. Primarna vrsta AI koja se koristi kod računalnog vida je ona koja se temelji na konvolucijskim neuronskim mrežama.



Slika 6.1 Primjer „računalnog vida“ [9]

Konvolucije su matematički procesi koje mreža koristi za određivanje razlike u vrijednostima između piksela. Ako se predvidi mreža vrijednosti piksela, potrebno je zamisliti manju rešetku kako se pomiče preko glavne mreže. Vrijednosti ispod druge rešetke analizira mreža, pa mreža istodobno ispituje samo nekoliko piksela. To se često naziva tehnikom "kliznih prozora". Vrijednosti koje analizira klizni prozor sažete su mrežom, što pomaže u smanjenju složenosti slike i olakšava mreži izdvajanje uzoraka. Konvolucijske neuronske mreže podijeljene su u dva različita dijela, konvolucijski dio i potpuno povezani dio. Konvolucijski slojevi mreže imaju ulogu izlučivanja, odnosno analiziranja piksela unutar slike i oblikovanje njihovih reprenzenata kako bi slojevi neuronske mreže mogu učiti uzorke iz njih. Konvolucijski slojevi počinju samo ispitivanjem piksela i ekstrahiranjem značajki slike na niskoj razini poput rubova. Kasniji slojevi spajaju rubove u složenije oblike. Na kraju mreža će prikazati rubove i detalje slike koje mogu prenijeti na potpuno povezane slojeve. Iako konvolucijska neuronska mreža može izvlačiti uzorke iz slika sama po sebi, točnost sustava računalnog vida može se uvelike poboljšati obilježavanjem slika. Obilježavanje slika (*engl. image annotation*) postupak je dodavanja metapodataka slici koji pomažu klasifikatoru u otkrivanju važnih objekata na slici. Upotreba obilježavanja za slike je važna kad god sustavi računalnog vida trebaju biti vrlo precizni, primjerice pri kontroli autonomnog vozila ili robota. Postoje različiti načini na koje se slike mogu obilježiti za poboljšanje performansi klasifikatora računalnog vida. Obilježavanje slika često se radi pomoću ograničavajućih okvira, okvira koji okružuje rubove ciljnog objekta i poručuje računalu da usredotoči svoju pažnju unutar okvira te će se upravo taj način obilježavanja slika koristiti u ovom projektu.

Semantička segmentacija je druga vrsta obilježavanja slika, koja djeluje dodjeljivanjem klase svakom pikselu na slici. Drugim riječima, svaki piksel koji bi se mogao smatrati "travom" ili "drvećem" biti će označen kao pripadnik tim klasama. Tehnika omogućuje preciznost na razini piksela, ali stvaranje semantičkih bilješki o segmentaciji je složenije i dugotrajnije od stvaranja jednostavnih okvira za ograničavanje [9].

6.2 Haar-Cascade klasifikator

Haar-Cascade je algoritam za automatsko otkrivanje objekata koji se koristi za identificiranje objekata na slici ili videozapisu, a zasnovan je na konceptu značajki koje su predložili Paul Viola i Michael Jones u svom radu " *Rapid Object Detection using a Boosted Cascade of Simple Features*" u 2001 [17]. To je pristup temeljen na strojnom učenju, gdje se kaskadna funkcija uvježbava iz mnogo pozitivnih i negativnih slika. Pozitivne slike su slike koje sadrže objekte koje želimo da naš klasifikator identificira. Negativne slike su slike svega ostalog, koje ne sadrže objekt koji želimo otkriti.

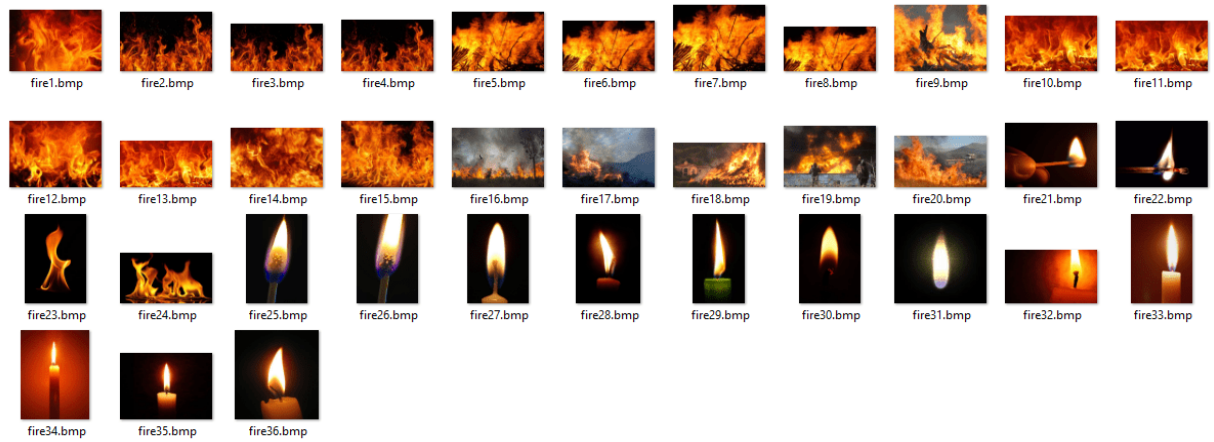
Haar-Cascade klasifikator poznat je po tome što može na slici otkriti lica i dijelove tijela, ali može se obučiti za prepoznavanje gotovo bilo kojeg predmeta [10].

6.3 Kreirarnje Haar-Cascade klasifikatora

Kreiranje Haar-Cascade datoteke izvesti ćemo koristeći besplatnu aplikaciju „dasar_haartrain“ preuzete sa stranice ovog projekta [11]. Izrada se sastoji od slijedećih koraka:

- odabir zbirke pozitivnih i negativnih slika
- označavanje pozitivnih slika pomoću alata objectmarker.exe
- stvaranje .vec (vektorske) datoteke na temelju pozitivnih označenih slika pomoću createsamples.exe
- obuka klasifikatora pomoću haartraining.exe
- pokretanje klasifikatora pomoću cvHaarDetectObjects ()

U ovom će se projektu za treniranje programa koristiti nekoliko pozitivnih i oko 100 negativnih uzoraka. Može se dodati više pozitivnih i negativnih slika snimljenih kamerom ili dodavanjem više javnih slika s internetskih izvora.



Slika 6.2 Zbirka pozitivnih slika za izradu klasifikatora [11]

Kao što smo već spomenuli, pozitivne slike su one slike koje sadrže objekt (npr. lice, oko ili bilo koji objekt kojeg želimo otkriti), u našem slučaju to je vatra, a negativne su one slike koje ne sadrže objekt od interesa (slike bez požara). Baza od više pozitivnih i negativnih slika uzrokuje veću preciznost klasifikatora.

Slika 6.3 Zbirka negativnih slika za izradu klasifikatora

Ovaj program sam sadrži nekakvih stotinjak negativnih slika, stoga neće biti potrebe za njihovim dodavanjem. Pozitivne slike pohranjujemo u direktorij: „dasar_haartrain/pozitive“ dok se negativne slike nalaze u direktoriju: „dasar_haartrain/negative“. U tom istom direktoriju negativnih slika nakon kompletiranja obiju zbirki potrebno je pokrenuti *batch* datoteku „create_list.bat“. Pokretanjem ove datoteke kreira se tekstualna datoteka s listom negativnih slika:

neg-0001.jpg

neg-0003.jpg

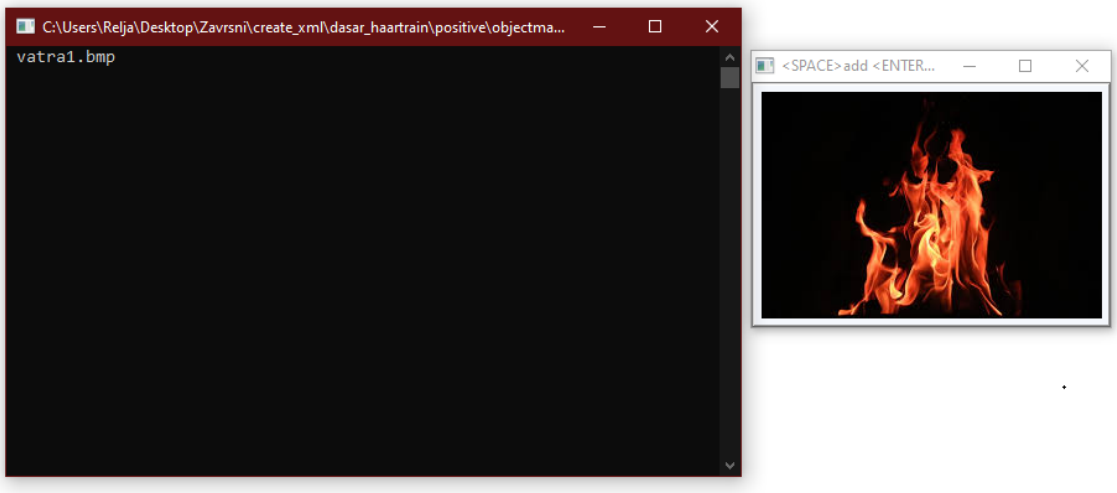
neg-0004.jpg

neg-0005.jpg

neg-0006.jpg…

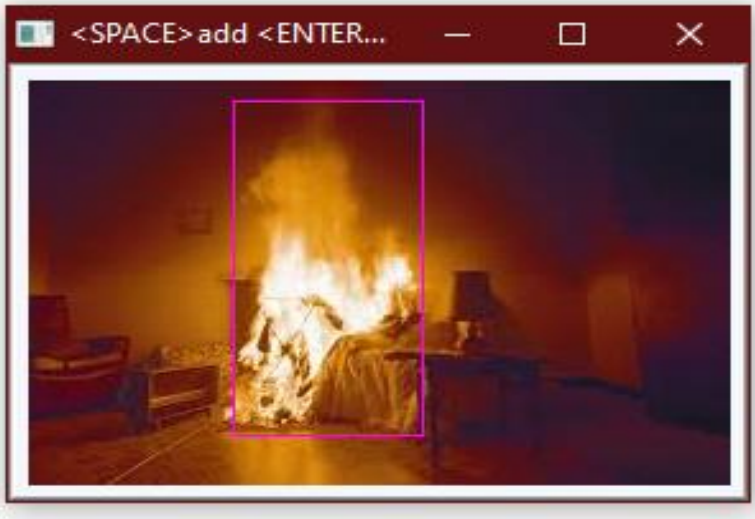
koja će se kasnije upotrijebiti pri treniranju klasifikatora. Nakon toga potrebno je označiti vatru na pozitivnim slikama kako bi programu dali do znanja na koje objekte treba obratiti pažnju pri svom radu. U ovom koraku stvara se datoteka podataka (*vektor file*) koja sadrži nazive pozitivnih slika kao i mjesto predmeta u svakoj slici. Ovu datoteku može se stvoriti pomoću dva alata: *Objectmaker* ili *Image Clipper*.

Prvi je jednostavniji i brži, a drugi alat je malo svestraniji, ali zahtjeva više vremena pri označavanju. Koristiti ćemo *Objectmaker* alat koji se nalazi u mapi „dasar_haartrain/pozitive/ objectmaker.exe“. Da bi se ispravno pokrenuo objectmaker.exe dvije datoteke cv.dll i highgui.dll također trebaju postojati u trenutnom direktoriju. Nakon pokretanja objectmaker.exe alata otvaraju se dva prozora prikazana na slici 6.4



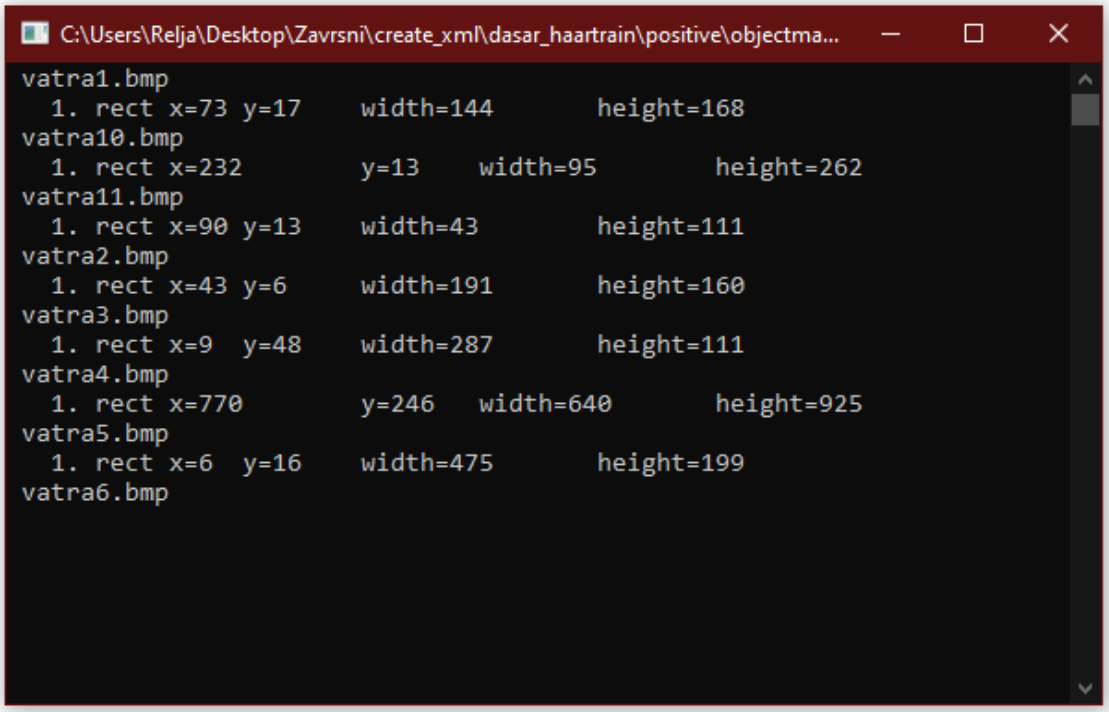
Slika 6.4 objectmaker.exe prozori za označavanje objekata na pozitivnim slikama

Zatim je potrebno na slici označiti područje od interesa, odnosno što preciznije označiti područje slike na kojem je prikazana vatra kao što je prikazano na slici 6.5. Klikom na gornji lijevi kut područja objekta i držanjem lijeve tipke miša povlačimo pravokutnik prema donjem desnom kutu objekta. Ukoliko smo zadovoljni označenim područjem tipkom „razmak“ a zatim „enter“ potvrđujemo odabir i prelazimo na slijedeću sliku. Ovaj proces ponavljamo za svaku pozitivnu fotografiju što može duže trajati ovisno o broju slika.



Slika 6.5 Označavanje požara na slici u Objectmaker alatu

Za svaku pozitivnu fotografiju alat će prikazati koordinate, širinu i visinu označenog područja iste, kao što je vidljivo na slici 6.6.



Slika 6.6 Prikaz parametara označenog područja

Alat će nakon završenog procesa označavanja kreirati tekstualnu datoteku imena info.txt sa slijedećim sadržajem:

rawdata/vatra1.bmp 1 70 26 141 154

rawdata/vatra10.bmp 1 240 19 89 244

rawdata/vatra11.bmp 1 80 22 48 98

rawdata/vatra2.bmp 1 69 14 155 152…

Prvi broj u svakom retku definira broj postojećih objekata na datoj slici. Na primjer, u drugom retku, broj 1 znači da je na slici vatra10.bmp odabran jedan objekt, a slijedeća četiri broja definiraju mjesto prvog objekta na slici (na svakoj slici moguće je označiti više objekata čije će brojke također biti navedene u istom formatu).

Nakon toga, potrebno je kreirati vektor pozitivnih slika pokretanjem *batch* datoteke „samples_creation.bat“. Navedena datoteka se sastoji od slijedećih bitnih parametara:

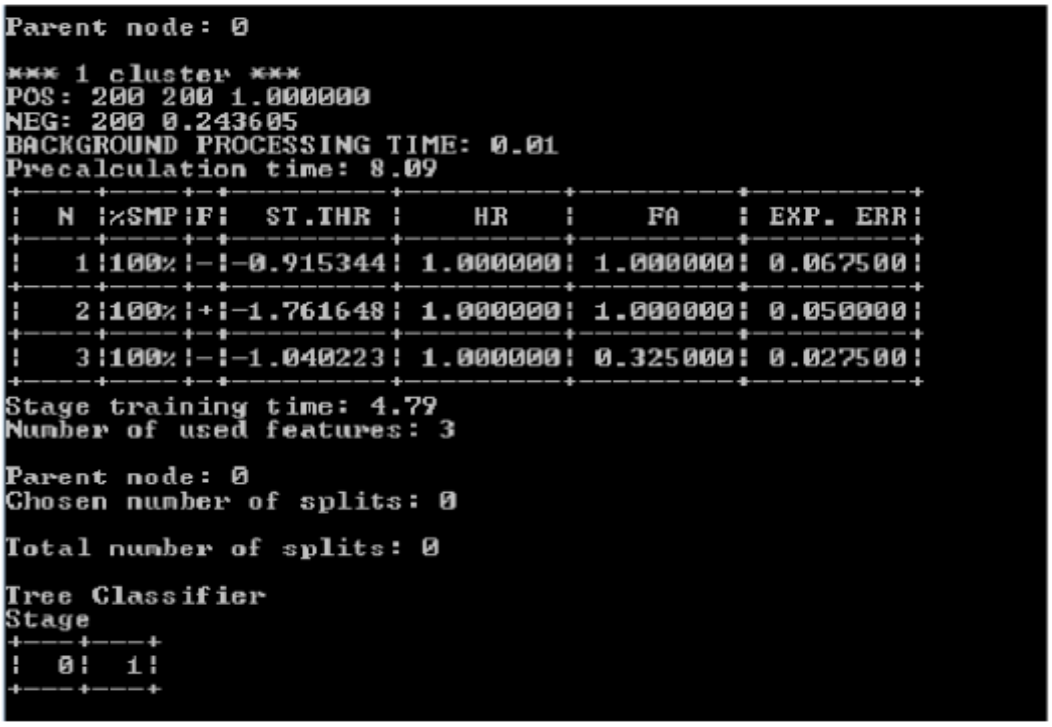
- info positive/info.txt – direktorij za pozitivnu info textualnu datoteku
- vec vector/facevector.vec - diektorij za izlaznu *vector* datoteku
- num 200 - broj pozitivnih datoteka koje će biti upakirane u vektor datoteku
- w 24 – širina objekata
- h 24 - visina objekata

Batch datoteka učitava info.txt i pakira objektne slike u vektorsku datoteku s imenom npr. facevector.vec. Nakon što se pokrene batch datoteka, imat ćemo datoteku facevector.vec u mapi: …\ dasar_haartrain\ vector. Zatim, potrebno je pokrenuti haartraining.exe datoteku koja se sastoji od slijedećih parametara:

- direktorija za spremanje islaine datoteke (XML datoteka: Haar-Cascade lasifikator)
- vec data/vector.vec: direktorija vektorske datoteke
- bg negative/bg.txt : direktorija za tekstualnu datoteku s listom negativnih slika
- npos 26: broj pozitivnih uzoraka ≤ broju pozitivnih slika
- nneg 100 broj negativnih uzoraka ≥ npos
- stade 15: broj predviđenih faza treninga
- mem 1024: količina memorije dodijeljena u MB
- w 24 -h 24: veličina uzorka
- nonsym: upotrebljava se ukoliko objekt nije vodoravno simetričan

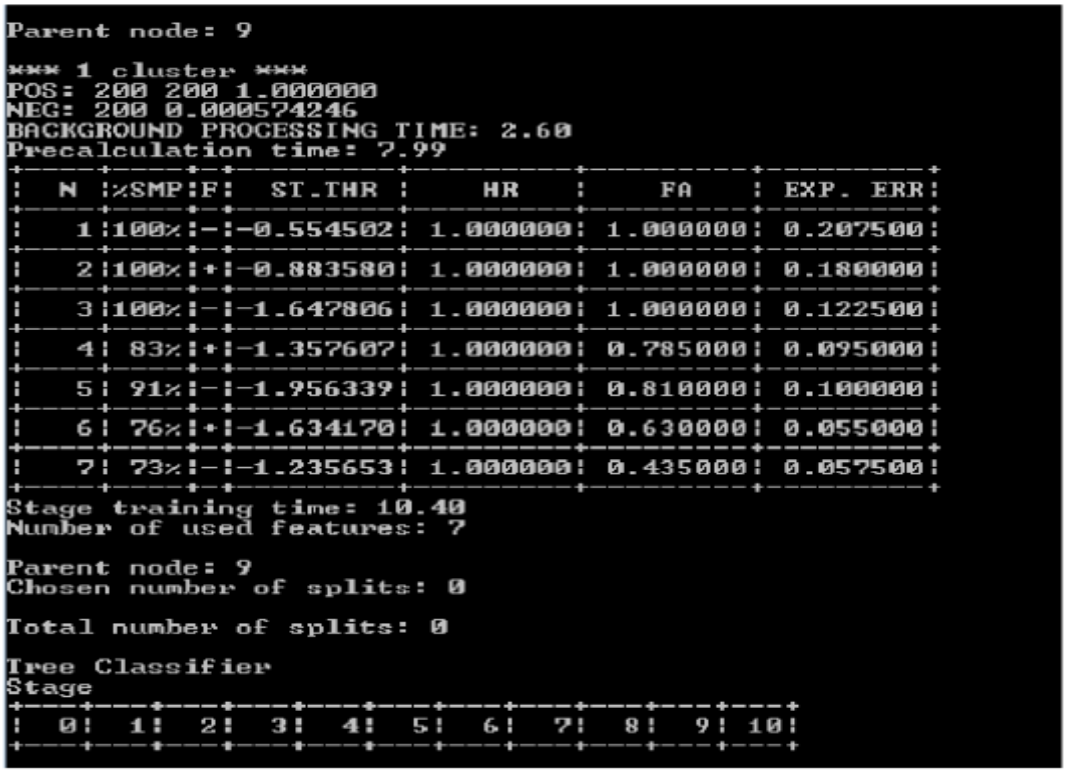
Haartraining.exe sakuplja novi set negativnih uzoraka za svaku fazu, nneg postavlja limit za veličinu seta. Haartraining.exe koristi podatke prethodnih faza da bi utvrdio koji su uzorci kandidati pogrešno klasificirani. Obuka se završava kada je ratifikacija pogrešno klasificiranih uzoraka niža od stupnja FA, bez obzira na broj stadija (nstages) koji su definirane u haartraining.bat,

program se može prekinuti ranije ukoliko dostignemo gornji uvjet što je obično dobar znak točnosti u našem procesu. Na slici 6.7 može se vidjeti izgled haartraining.exe prozora s pripadajućim parametrima.



Slika 6.7 haartraining .exe prozor; početak procesa

- Parent node: definira trenutnu fazu u procesu obuke
- N: broj korištenih značajki u ovoj fazi
- %SMP: postotak uzorka (postotak uzorka korišten za ovu značajku) F: "+" ukoliko je okrenut (y simetrija korištena) i "–" ukoliko nije
- ST.THR: prag faze
- HR: Hit Rate, na temelju praga faze
- FA: False Alarm, na temelju praga faze
- EXP. ERR: eksponencijalna pogreška klasifikatora



Slika 6.8 haartraining .exe prozor; kraj procesa

Sa slike 6.8 se može očitati da je za desetu fazu klasifikatora korišteno ukupno sedam značajki te je ukupan broj lažnih alarma opao a vrijeme računanja za obuku se povećalo. Za kraj, nakon odrađivanja ovog procesa u datoteci „cascades“ trebali bi biti sadržani katalozi imenovani od "0" do "N-1" gdje je N broj stupnjeva prethodno definiran u haartraining.bat. U svakom od tih kataloga trebala bi postojati datoteka AdaBoostCARTHaarClassifier.txt. Potrebno je kopirati sve mape 0..N-1 u mapu ../cascade2xml/data/. Naposljetku treba ukombinirati sve stvorene faze (klasifikatore) u jednu XML datoteku koja će biti konačna datoteka **Haar-Cascade klasifikator** i to na način da se pokrene convert.bat u direktoriju ../cascade2xml/. Kreiranu XML datoteku potom treba spremiti u odgovarajući direktorij kako bi je program Visual Basic mogao koristiti pri digitalnoj analizi slike.

7. KREIRANJE KORISNIČKE APLIKACIJE U VISUAL BASIC-U

U ovom poglavlju opisati će se Visual Basic programski alat, EmguCV platforma, kreiranje aplikacije za detekciju požara korak po korak te programski kod korišten za rad sustava.

7.1 Visual Basic 2010 expres edition

Microsoft Visual Studio je Microsoftovo integrirano razvojno okruženje (IDE). Koristi se za razvoj računalnih programa, kao i web stranica, web aplikacija, web usluga i mobilnih aplikacija. Visual Studio koristi Microsoftove platforme za razvoj softvera kao što su Windows API, Windows Forms, Foundation Presentation Foundation, Windows Store i Microsoft Silverlight. Može proizvesti i izvorni kod i upravljani kod. Visual Studio uključuje uređivač koda koji podržava *IntelliSense* (komponenta dovršetka koda) kao i kôd za preradu koda. Integrirani program za ispravljanje pogrešaka radi i kao ispravljač na razini izvora i kao ispravljač na razini stroja. Ostali ugrađeni alati uključuju kodni profil, dizajner za izgradnju GUI aplikacija, web dizajner, dizajner klasa i dizajner shema baza podataka. Prihvaća dodatke koji poboljšavaju funkcionalnost na gotovo svim razinama - uključujući dodavanje podrške za sustave upravljanja izvorima (poput Subervision-a i Git-a) i dodavanje novih skupova alata poput uređivača i vizualnih dizajnera za jezike specifične za domenu ili skupove alata za druge aspekte razvoja softvera. Visual Studio podržava 36 različitih programskih jezika i omogućuje uređivaču koda i programu za uklanjanje pogrešaka da podrže (u različitoj mjeri) gotovo bilo koji programski jezik pod uvjetom da postoji usluga specifična za jezik. Ugrađeni jezici uključuju C, C ++, C ++/CLI, Visual Basic .NET, C #, F #, JavaScript, TypeScript, XML, XSLT, HTML i CSS. Podrška za ostale jezike kao što su Python, Ruby, Node.js i M dostupna je putem dodataka. U prošlosti su podržani Java (i J #).

Visual Studio IDE pruža tri usluge: *SVsSolution*, koja pruža mogućnost nabrajanja projekata i rješenja; *SVsUIShell*, koji pruža funkciju prozora i korisničkog sučelja (uključujući kartice, alatne trake i prozore alata); i *SVsShell* koji se bavi registracijom *VSPackages-a*. Osim toga, IDE je također odgovoran za koordinaciju i omogućavanje komunikacije između servisa. Svi uređivači, dizajneri, vrste projekata i ostali alati implementirani su kao *VSPackages*. Visual Studio koristi COM za pristup *VSPackages-ima*. SDK Visual Studio također uključuje Okvir upravljanog paketa (MPF), koji je skup upravljanih omotača oko COM-sučelja koji omogućuju

pisanje paketa na bilo kojem jeziku koji je sukladan CLI-u. Međutim, MPF ne pruža svu funkcionalnost izloženu COM sučeljima Visual Studio. Usluge se tada mogu konzumirati za stvaranje drugih paketa koji dodaju funkcionalnost Visual Studiu.

IDE Visual Studio 2010 redizajniran je, prema Microsoft-u, tako da briše UI organizaciju i "smanjuje nered i složenost". Novi IDE bolje podržava više prozora dokumenata i plutajuće prozore alata, istovremeno nudeći bolji multi-monitor.

Visual Studio 2010 dolazi s .NET Framework 4 i podržava razvoj aplikacija koje ciljaju Windows 7. Podržava baze podataka IBM DB2 i Oracle, uz Microsoft SQL Server. Posjeduje integriranu podršku za razvoj Microsoft Silverlight aplikacija, uključujući interaktivni dizajn. Visual Studio 2010 nudi nekoliko alata za pojednostavljenje paralelnog programiranja: pored paralelnih proširenja za .NET Framework i biblioteke paralelnih uzoraka za izvorni kod, Visual Studio 2010 uključuje alate za uklanjanje pogrešaka paralelnih aplikacija. Novi alati omogućuju vizualizaciju paralelnih zadataka i njihovih vremena izvođenja. Alati za profiliranje paralelnih aplikacija mogu se koristiti za vizualizaciju čekanja niti i migracija niti kroz jezgre procesora [12].

Visual Studio 2010 je besplatan te zbog svojih mogućnosti predstavlja pogodno razvojno okruženje za ovaj projekt . Kao što je već i spomenuto sadrži razvojno okruženje za kreiranje GUI aplikacija kojeg ćemo koristiti pri izradi korisničkog sučelja.

7.2 Emgu CV

Emgu CV je .Net platforma (*wrapper*) koji omogućuje pristup do *OpenCV library-a* za obradu slika. Dopušta OpenCV [16] funkcijama da budu pozvane s .NET kompatibilnim jezicima. *Wrapper* moze biti kompajliran na Visual Studio-u, Xamarin Studio-u i Unity-u, a može se izvoditi na Windows, Linux, Mac OS X, iOS i Android operacijskim sustavima.

Emgu CV je napisan u cijelosti u C#. Prednost je u tome što se može sastaviti u Mono-u i zato se može pokretati na bilo kojoj platformi koju Mono podržava, uključujući iOS, Android, Windows Phone, Mac OS X i Linux. Puno je napora uloženo u čistu C # implementaciju jer se zaglavlja moraju prenijeti, u usporedbi s upravljanom C ++ implementacijom u koju se datoteke zaglavlja jednostavno mogu uključiti. Emgu CV može se koristiti s nekoliko različitih jezika, uključujući C#, VB.NET, C ++ i IronPython [13]. Druge prednosti Emgu CV- a su:

- klasa slike s općenitom bojom i dubinom
- automatski odvoz smeća
- XML dokumentacija i *IntelliSense* podrška
- mogućnost izravnog pozivanja iz OpenCV-a

OpenCV (*engl. Open Source Computer Vision Library*) je *open source library* za računalni vid i računalno strojno učenje. OpenCV je izgrađen kako bi pružio zajedničku infrastrukturu za aplikacije računalnog vida i ubrzao korištenje računalne percepcije u komercijalnim proizvodima. Budući da je licenciran BSD proizvod, OpenCV tvrtkama olakšava korištenje i izmjenu koda. Biblioteka posjeduje više od 2500 optimiziranih algoritama, koji uključuje sveobuhvatan skup klasičnih i vrhunskih algoritama računalnog vida i strojnog učenja. Ovi se algoritmi mogu upotrijebiti za otkrivanje i prepoznavanje lica, prepoznavanje objekata, klasificiranje ljudskih radnji u videozapisima, praćenje pokreta kamere, praćenje pokretnih objekata, izvlačenje 3D modela objekata, stvaranje 3D oblaka točaka iz stereo kamera, spajanje slika zajedno za stvaranje visoke rezolucije slika cijelog prizora, pronalaženje slične slike iz baze podataka, uklanjanje crvenih očiju sa slika snimljenih bljeskalicom, praćenje pokreta očiju, prepoznavanje krajolika itd. Procijenjen broj preuzimanja OpenCVa prelazi brojku od 18 milijuna. Uvelike se koristi u tvrtkama, istraživačkim grupama i vladinim tijelima [14].

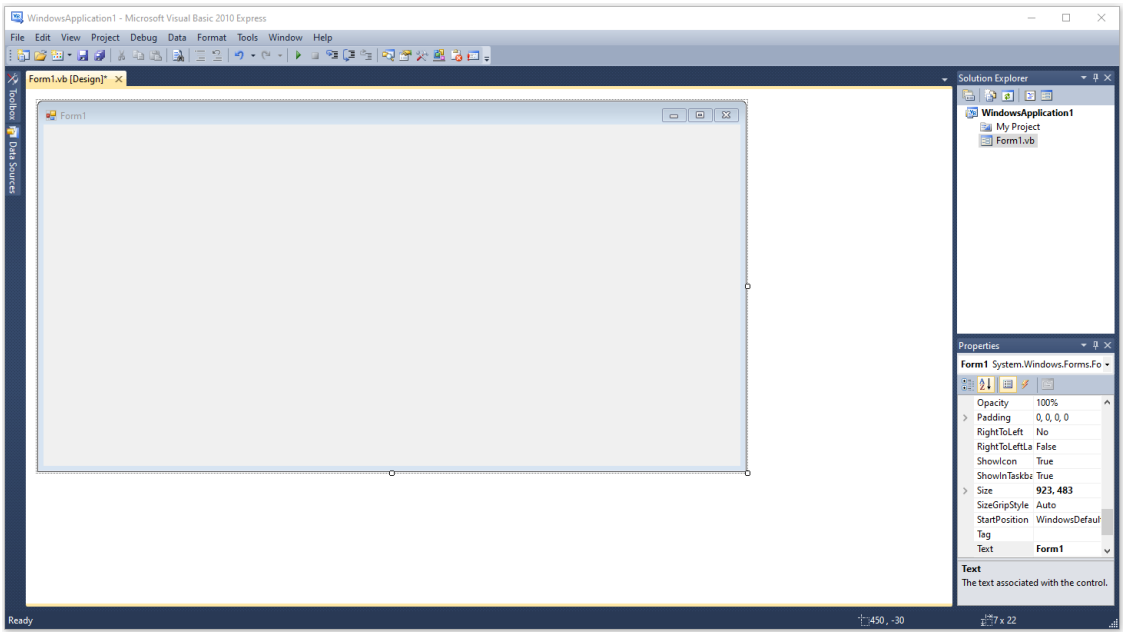
Korištenje Emgu CV-a omogućava također:

- korištenje različitih tipova boja (Gray, Bgr, Bgra. Hsv, Hls, Lab, Luv, Xyz, Ycc…)
- korištenje različitih tipova dubine boja (Byte, SByte, Single, Double, UInt16, Int16…)
- kreiranje slike
- učitavanje slike iz datoteke
- kreiranje slike iz Bitmap-a
- izvođenje osnovnih matematičkih operacija nad slikama
- crtanje objekata na slikama
- konverzija boje i dubine
- prikaz slike
- konverzija u Bitmap
- konverzija u XML
- konverzija iz XML-a [11]

7.3 Kreiranje forme

Za kreiranje korisničke aplikacije potrebna su nam dva alata spomenuta u ovom poglavlju, Microsoft Visual Basic 2010 express Edition i EmguCv instalacija.

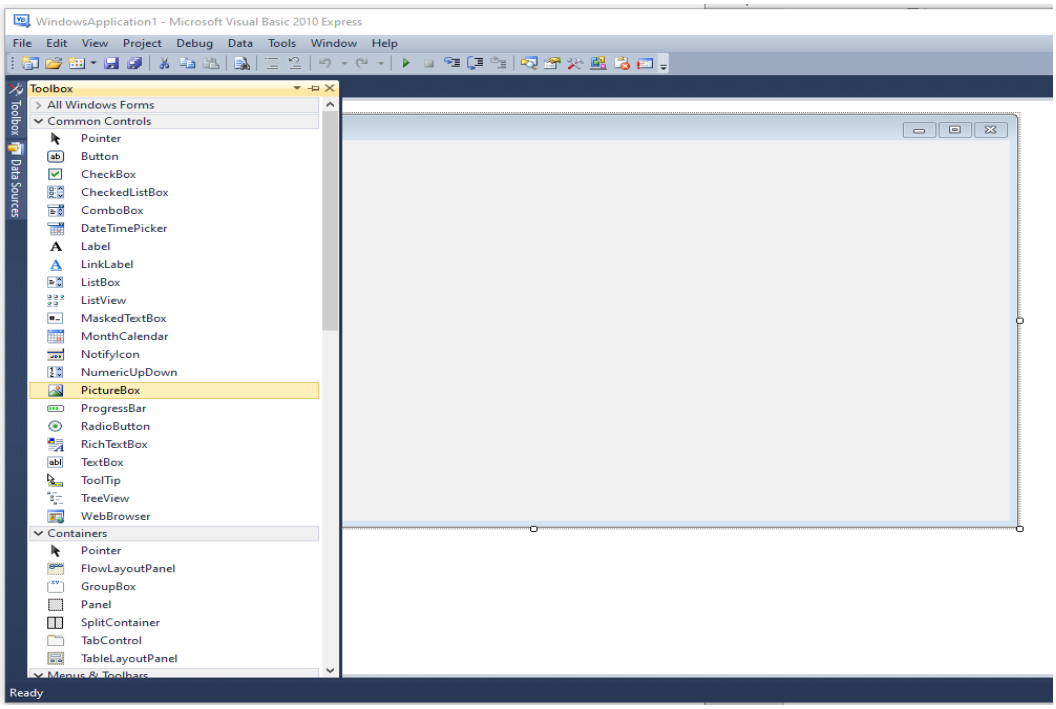
U Visual Basic-u pri stvaranju novog projekta potrebno je odabrati Windows Forms Application, odnosno projekt sa Windows korisničkim sučeljem kao na slici 7.1.



Slika 7.1 Windows Form Application

Windows korisničko sučelje omogućuje nam stvaranje aplikacije koju će korisnik upotrebljavati u ovom projektu. Aplikacija će prikazivati signal s web kamere u realnom vremenu te upozoravati korisnika o mogućem detektiranom požaru na slici, o porastu temperature u prostoru odnosno poslanom e-mail upozorenju ukoliko je požar zaista otkriven.

Da bi to bili u stanju realizirati potrebne su nam slijedeće stavke iz *Toolbox* kartice Visual Basic programa koje ćemo metodom „*drag and drop"* dodati u našu formu. Prva stavka je *PictureBox* kontrolna stavka koja će služiti za prikaz video signala s web kamere, zatim je potrebno dodati četiri oznake (*engl. Label*) od kojih Label1 prikazuje X koordinatu detektiranog objekta na slici, Label2 prikazuje Y koordinatu objekta, Label3 označava trenutak kada je poslano e-mail upozorenje za detektirani požar dok Label4 služi za ispis poruke da li je mogući požar na slici opažen ili ne. Ukoliko program detektira požar na slici prikazati će se poruka „POSSIBLE FIRE DETECTED!" u crvenoj boji, odnosno „No fire." u crnoj boji ako požara nema. Bitno je naglasiti da poruka glasi „mogući požar detektian" jer u nekim slučajevima program lažno prijavi požar na slici stoga je potrebno ispuniti i drugi uvjet, temperaturni porast, da bi požar zapravo bio otkriven a korisnik alarmiran.



Slika 7.2 Visual Basic Toolbox kartica

Nakon toga u formu dodajemo *TextBox* okvir koji inače omogućuje korisniku upis teksta, ali u našem slučaju koristiti ćemo ga za prikaz vrijednosti temperature koju program očitava preko Arduino sučelja povezanog na serijskom ulazu računala. Ukoliko je očitana vrijednost temperature veća od postavljene temperaturne granice, program uz iznos temperature ispisuje poruku „High Temperature“, a inače dok je temperatura prostora normalna program ispisuje poruku „Normal Temperature“. U formu je još potrebno dodati i *CheckBox* koji će nam označavati trenutak kada program očita visoku temperaturu sa TextBox okvira. Visoka temperatura (poruka „High Temperature“ u tekstualnom okviru) *CheckBox1* postavlja u stanje *true* i ispisuje poruku „TEMPERATURE INCREASE!“ u crvenoj boji. Normalna temperatura prostora (poruka „Normal Temperature“ u tekstualnom okviru) *CheckBox1* postavlja u *false* stanje i prikazuje poruku „No fire“ u crnoj boji.

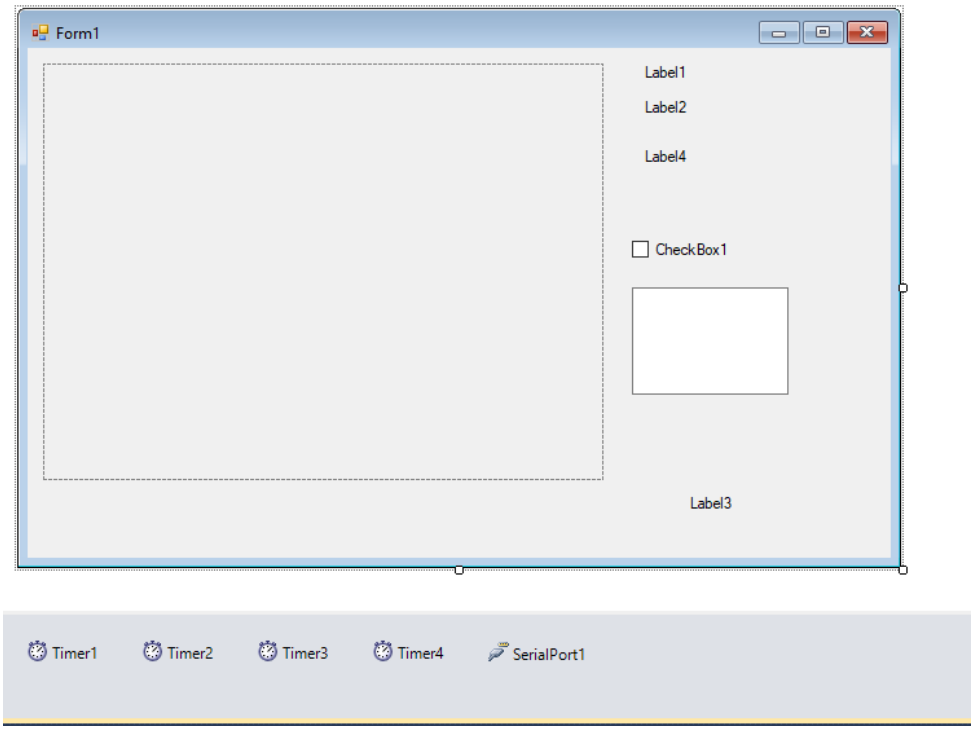
Slijedeće komponente koje su nam potrebne su brojači i serijski port. Brojač je ugrađena VB.Net kontrola koja nam omogućuje izvršavanje koda nakon što protekne određeno vrijeme od trenutka kada je tajmer omogućen ili više puta u određenim vremenskim intervalima [15]. Jednom omogućeno, tajmer će generirati *Tick* događaj u unaprijed određenim intervalima. Duljina intervala određena je u milisekundama. Zadani interval je 100 milisekundi, ali raspon mogućih vrijednosti je 1 ms do 2 147 483 647 ms.

Koristiti ćemo četiri brojača koji nam omogućuju da izvršimo nekakve radnje u određenim intervalima pa stoga definiramo:

- ***Timer1***: služi za definiranje vremenskog intervala u kojem će program izvršiti funkciju za detekciju požara (funkciju *fire()),* svojstvo *Enabled* postavljamo u stanje *True,* a vemenski interval na 100ms
- ***Timer2***: definira vremenski interval u kojem će program provjeravati da li su oba uvjeta za detekciju požara ispunjena. Ukoliko jesu, program će generirati e-mail upozorenje za korisnika i poslati poruku na njegovu e-mail adresu. Svojstvo *Enabled* postavljamo u stanje *True,* a vemenski interval na 100ms
- ***Timer3***: koristimo za određivanje vremenskog intervala pri kojem će program brisati sadržaj tekstualnog okvira kako bi nove vrijednosti očitane sa temperaturnog senzora bile prikazane prve. Bez ovog brojača samo bi par prvih očitanih vrijednosti bilo moguće pročitati sa aplikacije, dok one svježije ne bi bile vidljive. Svojstvo *Enabled* također postavljamo u stanje *True,* a vemenski interval na 4000ms

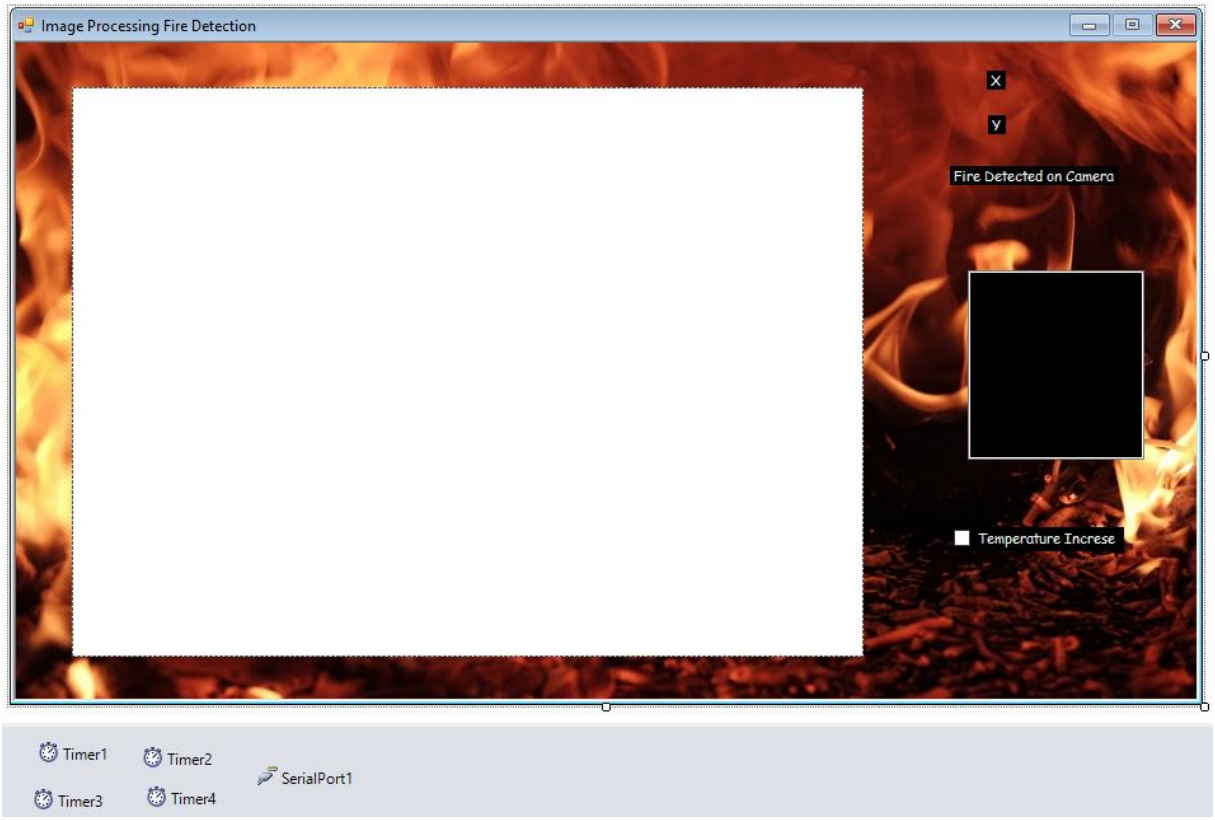
- ***Timer4***: služi za definiranje intervala pri kojem će program provjeravati da li je očitana vrijednost temperature prešla graničnu vrijednost, odnosno da li je sadržaj tekstualnog okvira poruka „High Temperature“ ili „Normal Temperature“. Svojstvo *Enabled* također postavljamo u stanje *True*, a vemenski interval na 1000ms.

Serijski port (*SerialPort1*) potrebno je uključiti u aplikaciju kako bi mogli ostvariti komunikaciju između Arduino sučelja i računalnog programa. Bez serijskog porta ne bi bili u mogućnosti prikazivati vrijednosti temperature sa temperaturnog senzora u korisničkoj aplikaciji. Sva zadana svojstva porta nam odgovaraju pa ih nećemo mijenjati osim svojstva *PortName*, za ime serijskog ulaza upisati ćemo „COM3“ jer je taj serijski ulaz korišten pri komunikaciji sa Arduinom.



Slika 7.3 Kreirano korisničko sučelje

Na slici 7.3 je prikazano korisničko sučelje sa svim dodanim elementima. Nakon još malo dorade aplikacije u smislu dodavanja pozadinske slike, imena oznaka i drugih elemenata te podešavanju veličine i rasporeda komponenti dobivamo konačni izgled aplikacije prikazan na slici 7.4



Slika 7.4 Konačan izgled aplikacije za detekciju požara

7.4 Programski kod

Nakon što smo kreirali korisničku aplikaciju za detekciju požara potrebno je programirati način na koji će aplikacija raditi i prikazivati podatke. Prije samog programiranja potrebno je u projekt dodati prethodno preuzete EmguCv datoteke kako bi program imao pristup istima. Na slijedećim stranicama prikazan je detaljno opisan programski kod.

```
Imports System.Runtime.InteropServices 'System.Runtime.InteropServices
omogućava pristup zbirci klasa korisnih za pristup COM objektima
Imports System.Threading 'System.Threading pruža klase i sučelja koja
omogućuju višeslojno programiranje
Imports System.IO 'System.IO sadrži zbirku koje omogućuju čitanje i pisanje
u datoteke
Imports System.Net ' Ova zbirka nudi dostupne opcije za postavke proxy-ja
koje HttpClient koristi prilikom pokretanja u sustavu Windows
Imports System.Net.Mail ' ova zbirka pruža opcije za slanje e-mail poruke

Imports System.Net.NetworkCredential ' NetworkCredential klasa je osnovna
klasa koja isporučuje vjerodajnice u shemama provjere autentičnosti na
temelju lozinke
Imports System.Reflection ' služi za dohvaćanje informacije o sklopovima,
modulima, članovima, parametrima i drugim entitetima u upravljanom kodu
ispitivanjem njihovih metapodataka.
Imports System.Text 'sadrži klase koje predstavljaju kodiranje ASCII i
Unicode znakova; apstraktne osnovne klase za pretvaranje blokova znakova u
i iz blokova bajtova; i pomoćna klasa koja manipulira i formatira String
objekte bez stvaranja posrednih instanci Stringa
Imports Microsoft.Win32 ' pruža dvije vrste klasa: one koje obrađuju
događaje pokrenute u operativnom sustavu i one koje manipuliraju sistemskim
registrom.

Imports System.IO.Ports 'System.IO.Ports sadrži klase za kontrolu serijskih
portova

Imports Emgu.CV
Imports Emgu.CV.Util

Imports Emgu.CV.Structure 'EmguCV detaljno je opisan u poglavlju 7.2

Imports System.Diagnostics ' System.Diagnosticsnudi klase koje omogućuju
interakciju s sistemskim procesima, zapisima događaja i brojačima
performansi


Public Class Form1 'deklaracija klase Form1

    Dim firedetected As Integer 'deklaracija integer objekta firedetected,
ova varijabla u daljnjem programu poprima vrijednost 0 ukoliko požara nema
na slici odnosno 1 ukoliko je požar otkriven

    Dim web As Capture = New Capture 'deklariranje objekta tipa „capture“,
ovaj objekt će nam omogućavati prikaz video signala sa web kamere


    Private Declare Function InternetGetConnectedState Lib "wininet.dll"
(ByRef lpdwFlags As Int32, ByVal dwReserved As Int32) As Boolean
'deklaracija funkcije „InternetGetConnectedState“ tipa bool. Ova funkcija
će nam u daljnjem kodu koristiti za provjeru internetske veze


    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick 'postavljanje vremenskog intervala za
brojač Timer1 u kojem ćemo izvršavati funkciju za detekciju požara
„fire()“. Pri kreiranju korisničke aplikacije ovaj interval smo postavili
na 100mS

      fire()
End Sub
```

```
    Sub fire()'funkcija fire() za detekciju požara

        Dim photo As Image(Of Bgr, Byte)'deklaracija objekta „photo" tipa
image (slika) s parametrima: tip boje Bgr, tip dubine: Byte

        Dim currentCenter As New Point()'deklariranje objekta
„currentCenter" tipa Point, koristiti ćemo ga pri određivanju koordinata
detektiranog požara u daljnjem kodu

        photo = web.RetrieveBgrFrame 'objektu „photo" pridodajemo
vrijednost objekta „web", odnosno video signala sa web kamere

        Dim objectdetection As New
CascadeClassifier("C:\data\haarcascades\fire.xml") ' deklaracija objekta
„obectdetection" kao Haar-Cascade klasifikatora s direktorijem koji ukazuje
na sašu kreiranu XML datoeku za detekciju požara


     Try ' pokušaj izvršiti slijedeći dio koda
         Dim image As Image(Of Gray, Byte) = photo.Convert(Of Gray,
Byte)()' deklaracija objekta „image" tipa image s parametrima: tip boje:
Gray, tip dubine: Byte; objektu „image" pridjeljujemo vrijednost objekta
„photo" konvertiranog u sivu sliku ( tip boje Gray, tip dubine Byte )

For Each FIRE As Rectangle In objectdetection.DetectMultiScale(image, 1.1,
8, Size.Empty, Size.Empty) ' za svaki objekt detektiran na slici „image"
kreiraj pravokutnik veličine 1.1, 8 (ostale dvije točke koje definiraju
pravokutnik su postavljene kao „Size.Empty" kako bi se veličina
pravokutnika prilagodila veličini detektiranog objekta na slici)

 photo.Draw(FIRE, New Bgr(Color.Red), 2)' boja pravokutnika crvena,debljine
2

           currentCenter.X = CInt(CDbl(FIRE.X + FIRE.X + FIRE.Width) /
2.0)'izračun trenutne X koordinate tetektiranog objekta

           currentCenter.Y = CInt(CDbl(FIRE.Y + FIRE.Y + FIRE.Height) /
2.0)' izračun trenutne Y koordinate detektiranog objekta

                 Label1.Text = (currentCenter.X).ToString ' u Label1 na
korisničkoj aplikaciji upiši trenutnu vrijednost X koordinate kao string

                 Label2.Text = (currentCenter.Y).ToString ' u Label na
korisničkoj aplikaciji upiši trenutnu vrijednost Y koordinate kao string


                 firedetected = 1   ' spremamo broj 1 u firedetected objekt
ako je požar otkriven na slici

             Next

             PictureBox1.Image = photo.ToBitmap ' u PictureBox1 prikaži
objekt photo (signal s web kamere) u Bitmap formatu

        Catch ex As Exception

            MessageBox.Show(ex.Message) ' ukoliko program naiđe na pogrešku
pri izvođenju gore navedenog koda, prikaži poruku pogreške
        End Try
```

```
        If firedetected = 1 Then
            Label4.Text = " POSSIBLE FIRE DETECTED!"
            Label4.BackColor = Color.Red ' ukoliko je požar detektiran
(firedetected = 1), tada u Label4 upiši tekst " POSSIBLE FIRE DETECTED!" i
postavi pozadinsku boju u crveno

        Else
            Label4.Text = " No fire."
            Label4.BackColor = Color.Black
            firedetected = 0 ' ukoliko požar nije detektiran (firedetected
≠ 0) upiši u Label4 tekst „No fire.“ I postavi pozadinsku boju u crno;
postavi firedetected u nulu

        End If
            End Sub

Private Sub Timer4_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer4.Tick ' postavi Timer4 na interval (1000mS
u korisničkoj aplikaciji) za izvršavanje slijedećeg koda

        If InStr(TextBox1.Text, "High Temperature") Then

            CheckBox1.Checked = True
            CheckBox1.Text = "TEMPERATURE INCREASE!"
            CheckBox1.BackColor = Color.Red
            TextBox1.BackColor = Color.Red ' ukoliko je sadržaj tekstualnog
okvira TextBox1 tekst „High Temperature“ (odnosno Arduino program je
detektirao povećanje temperature u prostoru sa temperaturnog senzora) tada
postavi CheckBox1 u stanje true, u CheckBox1 upiši tekst „TEMPERATURE
INCREASE!“, postavi pozadinsku boju u crveno, a pozadinsku boju tekstualnog
okvira TextBox1 postavi također u crveno
        End If

        If InStr(TextBox1.Text, "Normal Temperature") Then

            CheckBox1.Checked = False
            CheckBox1.Text = "No Fire."
            CheckBox1.BackColor = Color.Black
            TextBox1.BackColor = Color.Black ' ukoliko je sadržaj
tekstualnog okvira TextBox1 tekst „Normal Temperature“ (odnosno Arduino
program nije detektirao povećanje temperature u prostoru sa temperaturnog
senzora) tada postavi CheckBox1 u stanje false, u CheckBox1 upiši tekst „No
Fire.“, postavi pozadinsku boju u crno, a pozadinsku boju tekstualnog
okvira TextBox1 postavi također u crno

        End If

    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick ' postavi Timer2 na interval (100mS u
korisničkoj aplikaciji) za izvršavanje slijedećeg koda

        If ((CheckBox1.Checked = True) And (firedetected = 1)) Then '
ukoliko je ChecBox1, u stanju true i objekt firedetected = 1 tada izvrši
slijedeći dio koda; ovaj dio koda provjerava da li su oba uvjeta za
detekciju požara zadovoljena, porast temperature  u prostoru i prepoznat
požarni objekt na slici web kamere, kako bi program poslao upozorenje
korisniku
```

```
            Dim smtp As New SmtpClient
            Dim message As New MailMessage
            Dim username As String
            Dim password As String
            Dim destination As String
            Dim subject As String
            Dim lngFlags As Integer ' deklaracija objekata potrebnih za
slanje e-mail upozorenja korisniku kao što su korisničko ime (email
adresa), zaporka, odredište poruke upozorenja, naslova poruke…

            username = "detekcija.pozara@gmail.com"
            password = "detekcijapozara1994"
            destination = "franerelja@gmail.com"
            subject = "Fire Detected"
            smtp.Host = "smtp.gmail.com"
            smtp.EnableSsl = True
            smtp.Port = 587
            smtp.Credentials = New Net.NetworkCredential(username,
password) ' upisivanje podataka za email uzbunjivanje

            message.To.Add(destination)' pošalji poruku na odredište
„destination = "franerelja@gmail.com"“
            message.From = New MailAddress(username)' poruku šalje username
= "detekcija.pozara@gmail.com"

            message.Subject = subject ' naslov poruke subject = "Fire
Detected"
            message.Body = TextBox1.Text ' tekst poruke je sadržaj
tekstualnog okvira TextBox1 u trenutku detekcije (tekst se sastoji od
iznosa temperature očitanog u trenutku detekcije i poruke „High
Temperature“)
            If InternetGetConnectedState(lngFlags, 0) Then ' provjera
internetske veze, ukoliko je internetska veza uspostavljena tada izvrši
slijedeći dio koda

                smtp.Send(message)' pošalji email poruku
                TextBox1.Text = "" ' očisti sadržaj tekstualnog okvira
TextBox1
                CheckBox1.Checked = False ' postavi CheckBox1 u stanje
false
                Label3.Text = "ALERT SENT" ' upiši u Label3 u korisničkoj
aplikaciji tekst „AlERT SENT“ kako bi korisnik znao da je poslano
upozorenje


            Else
                TextBox1.Text = TextBox1.Text + "" + vbNewLine + " net
connection not available" + vbNewLine + "" ' ukolko internetska veza nije
uspostavljena tada u tekstualni okvir korisničke aplikacije upiši tekst
„net connection not available"“

           End If


        End If
    End Sub

   Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SerialPort1.Close()
        SerialPort1.PortName = "COM3"
```

```
        SerialPort1.BaudRate = "9600"
        SerialPort1.DataBits = 8
        SerialPort1.Parity = Parity.None
        SerialPort1.StopBits = StopBits.One
        SerialPort1.Handshake = Handshake.None
        SerialPort1.Encoding = System.Text.Encoding.Default
        SerialPort1.Open()' postavljanje parametara komunikacije s
serijskim portom SerialPort1 (Arduino sučeljem) kao što je ime serijskog
porta, paritet, baud brzina, broj bitova podataka, itd. Serijski port  prvo
zatvaramo, postavljamo navedene parametre a zatim ga otvaramo. Potrebno je
navedeno izvršiti u takvom slijedu zbog sinkronizacije serijskog porta i
Arduino sučelja

    End Sub


    Private Sub DataReceived(ByVal sender As Object, ByVal e As
SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived 'kreiranje
procedure „DataReceived" za primanje podataka sa serijskog porta
SerialPort1

        Try ' pokušaj izvršiti slijedeći dio koda

            Dim mydata As String = "" ' deklariranje objekta „mydata" kao
      prazan string

            mydata = SerialPort1.ReadExisting()' upiši u mydata objekt
očitane podatke sa serijskog porta SerialPort1

            If TextBox1.InvokeRequired Then
                TextBox1.Invoke(DirectCast(Sub() TextBox1.Text &= mydata,
MethodInvoker)) ' ukoliko je za tekstualni okvir TexBox1 potreban priziv
(invoke) tada u tekstualni okvir upiši vrijednost mydata objekta (podatke
sa serijskog porta) „invoker" metodom; ovaj način se koristi kada postoje
različite sekvence za izvođenje koda
            Else
                TextBox1.Text &= mydata ' upiši u tekstualni okvir
vrijednost mydata (podaci sa serijskog porta)
            End If

        Catch ex As Exception

            MessageBox.Show(ex.Message)  'ukoliko program naiđe na pogrešku
prilikom izvršavanja gornjeg koda prikaži poruku pogreške
        End Try


    End Sub

    Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
        TextBox1.Text = "" ' postavljanje vremenskog intervala za brojač
Timer3 (4000mS u formi Form1)za koji će u tekst tekstualnog okvira TextBox1
upisati „", odnosno očistiti prošle vrijednosti temperature očitane sa
senzora

    End Sub

End Class
```

8. ARDUINO

U ovom poglavlju opisati će se Arduino IDE programsko okruženje. Kao što je već i spomenuto, što se tiče Arduino dijela sustava za detekciju požara ideja je prilično jednostavna za spajanje i uporabu kako hardverski tako i softverski.

8.1 Arduino IDE

Arduino IDE je softver otvorenog koda koji se uglavnom koristi za pisanje i učitavanje koda u Arduino modul. Arduinovo integrirano razvojno okruženje (IDE) je aplikacija na više platformi (za Windows, macOS, Linux) koja je napisana u funkcijama iz C i C ++-a. Koristi se za pisanje i prijenos programa na kompatibilne Arduino pločice, ali i uz pomoć drugih jezgara, drugih razvojnih ploča drugih proizvođača. IDE okruženje uglavnom sadrži dva osnovna dijela: Editor i Compiler gdje se prvi koristi za pisanje potrebnog koda, a drugi dio se koristi za sastavljanje i učitavanje koda u zadani Arduino modul.

Izvorni kod za IDE objavljuje se pod GNU General Public License, verzija 2. Arduino IDE podržava jezike C i C ++ koristeći posebna pravila strukturiranja koda. Arduino IDE isporučuje softversku biblioteku iz projekta ožičenja, koji pruža brojne uobičajene postupke unosa i izlaza. Korisnički napisani kôd zahtijeva samo dvije osnovne funkcije za pokretanje skice i glavnu petlju programa, koje se sastavljaju i povezuju s programskim stub-om main () u izvršni ciklički izvršni program s alatnim lancem GNU-a, koji je također uključen u distribuciju IDE-a. Arduino IDE koristi programski „avrdude“ (mogućnost manipuliranja ROM i EEPROM sadržajem mikrokontrolera) za pretvaranje izvršnog koda u tekstualnu datoteku u heksadecimalnom kodiranju koji se u Arduino ploču učitava programom za učitavanje u firmware-u ploče. Standardno, „avrdude“ se koristi kao alat za učitavanje kako bi se korisnički kôd učitao na službenim Arduino pločicama.

S porastom popularnosti Arduino-a kao softverske platforme, drugi dobavljači počeli su primjenjivati prilagođene prevoditelje otvorenog koda i alate (jezgre) koji mogu izrađivati i prenositi skice u druge MCU-ove koji nisu podržani u službenoj liniji MC-ova Arduino-a. U listopadu 2019. organizacija Arduino započela je rani pristup novom Arduino Pro IDE-u s otklanjanjem pogrešaka i drugim naprednim značajkama.

Neke od prednosti Arduino integriranog razvojnog okruženja su: razvijen u tzv. „Ready to Use“ strukturi, sadrži zbirku primjera korisnih za rad i učenje, mnogobrojne funkcije olakšavaju rad te rasprostranjena upotreba od strane inženjera, studenata, profesionalaca i hobi korisnika koja omogućuje pronalazak odgovora za svaki problem pri radu [16].

8.2 Programski kod

```
#include <Wire.h> // zbirka funkcija koja omogućuje komunikaciju sa I2C/TWI uređajima kao što je BMP180 senzor

#include <SFE_BMP180.h>  // zbirka funkcija za komunikaciju s BMP180 senzorom

SFE_BMP180 bmp180;

void setup() // funkcija „setup()" se poziva samo jednom pri izvršenju skripte

{

  Serial.begin(9600); // baud brzinu postavljamo na 9600


bool success = bmp180.begin(); // deklaracija varijable „success" tipa bool kojoj pridjeljujemo vrijednost inicijalizacije rada BMP180 senzora


  if (success)

{

    Serial.println("BMP180 init success");

} // ukoliko je inicijalizacija senzora bila uspješna ispiši poruku „BMP180 init success" na serijskom portu računala


void loop() // funkcija „loop()" poziva se uvijek iznova i izvršava ispod navedeni dio koda

{

 char status; // deklaracija lokalne varijable „status" tipa char koju koristimo za provjeru statusa senzora pri očitavanju temperature

  double T; // varijabla „T" u koju ćemo spremiti očitanu vrijednost temperature

  bool success = false; //postavljamo varijablu „success" u stanje false

 status = bmp180.startTemperature(); //započinjemo očitavanje temperature sa senzora

if (status != 0) {

    delay(1000);

    status = bmp180.getTemperature(T); // ukoliko je vrijednost varijable status različita od nule, pričekaj jednu sekundu i očitaj trenutnu vrijednost temperature sa senzora i spremi je u varijablu T


          Serial.print("Temperature: ");

          Serial.print(T);
```

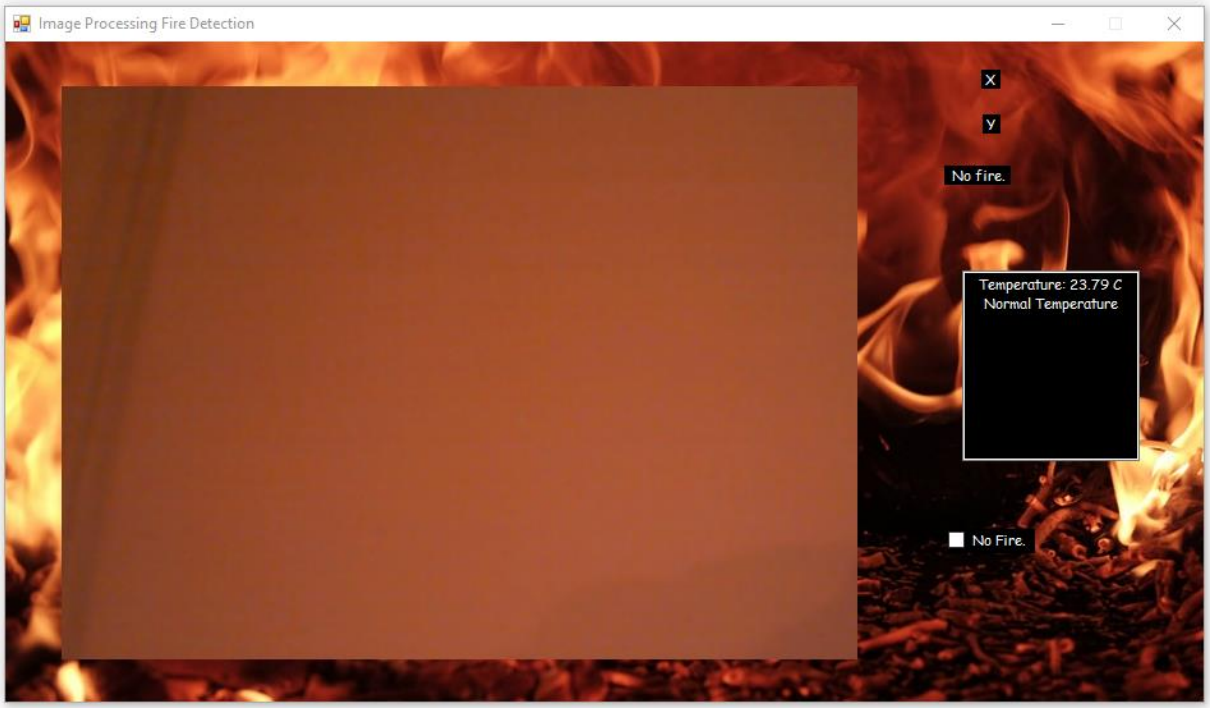
```
          Serial.println(" C"); //ispiši vrijednost temperature uz navedeni
tekst
           if(T < 30.90)
         {
           Serial.println("Normal Temperature");
         delay(1000); // pričekaj 1 sekundu
         }
// ukoliko je vrijednost temperature manja od 30.90 stupnjeva celzijevih
tada ispiši na serijski port poruku „Normal Temperature“. Ova vrijednost se
može podešavati ovisno o prostoru u kojem se nalazi senzor, odnosno prema
uobičajenom iznosi temperature za tu prostoriju


         if(T > 31.00)
         {
           Serial.println("High Temperature");
            delay(1000); // pričekaj 1 sekundu
// ukoliko je vrijednost temperature veća od 31.00 stupanj celzijevi tada
ispiši na serijski port poruku „High Temperature“. Ovu vrijednost također
podešavamo ovisno o prostoru u kojem se nalazi senzor, odnosno prema
uobičajenom iznosi temperature za tu prostoriju. U ovom slučaju za potrebe
rada smo postavili graničnu temperaturu nešto niže kako bi sustav prije
reagirao


         } else
         delay(500); // pričekaj pola sekunde
         }
       }
```

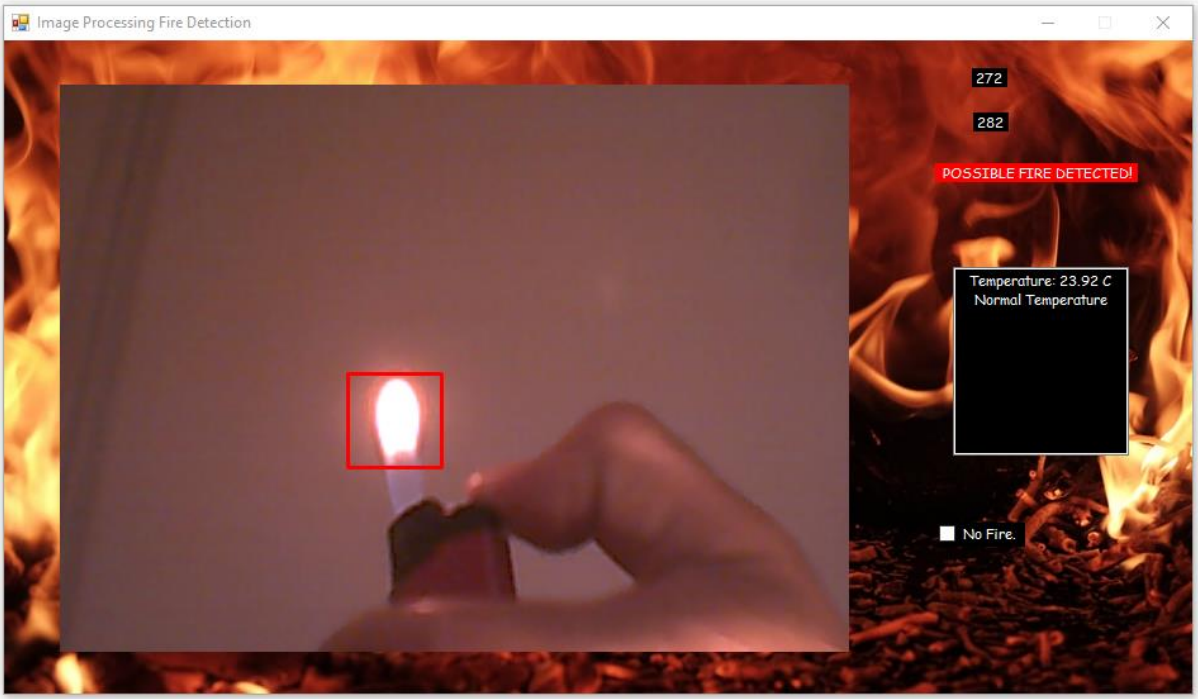
9. RAD SUSTAVA U STVARNIM UVJETIMA

Sustav za detekciju požara postavili smo u dnevnu sobu gdje očekujemo normalnu sobnu temperaturu u iznosu od 20 °C – 25 °C stoga smo, kako je i objašnjeno u Arduino kodu, postavili graničnu temperaturu na 31.00 °C kako bi sustav što prije reagirao na porast temperature.



Slika 9.1 Korisničko sučelje za detekciju požara u početnim uvjetima

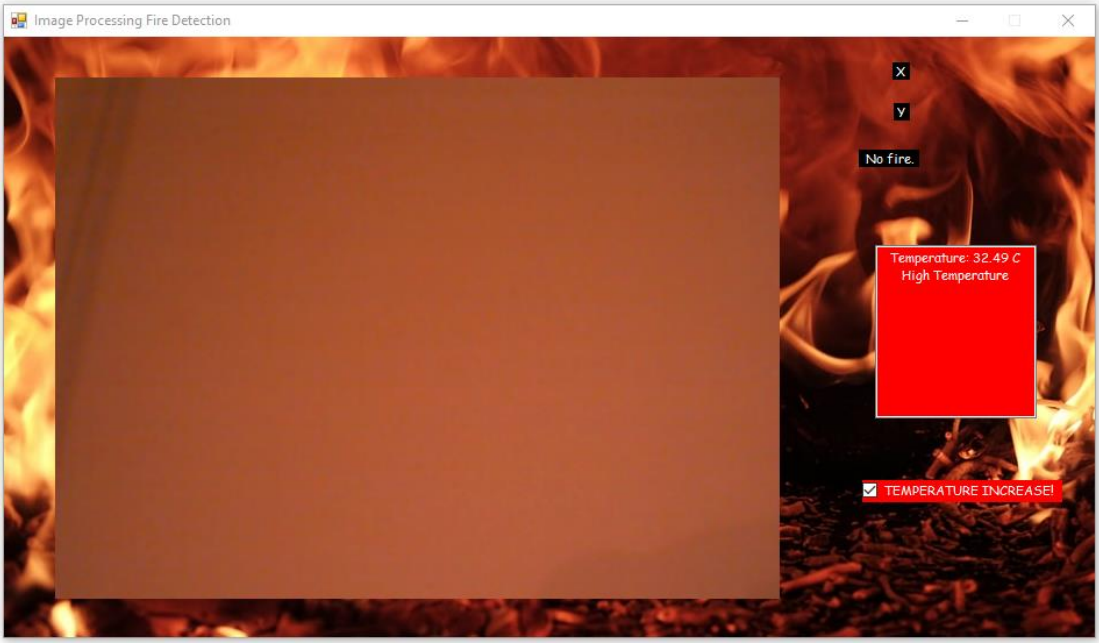
Kao što je i vidljivo na slici 9.1 sustav očitava sobnu temperaturu u iznosu od 23.79 °C i prikazuje signal s web kamere u realnom vremenu. Temperatura je u razini normale stoga sustav prikazuje poruku „Normal Temperature“ uz iznos temperature u tekstualnom okviru te je CheckBox1 u stanju *false* s tekstom „No Fire“. Koordinate objekta nisu prikazane jer program nije detektirao požarni objekt na slici stoga i Label4 prikazuje poruku „No fire“.



Slika 9.2 Detektirani objekt na slici

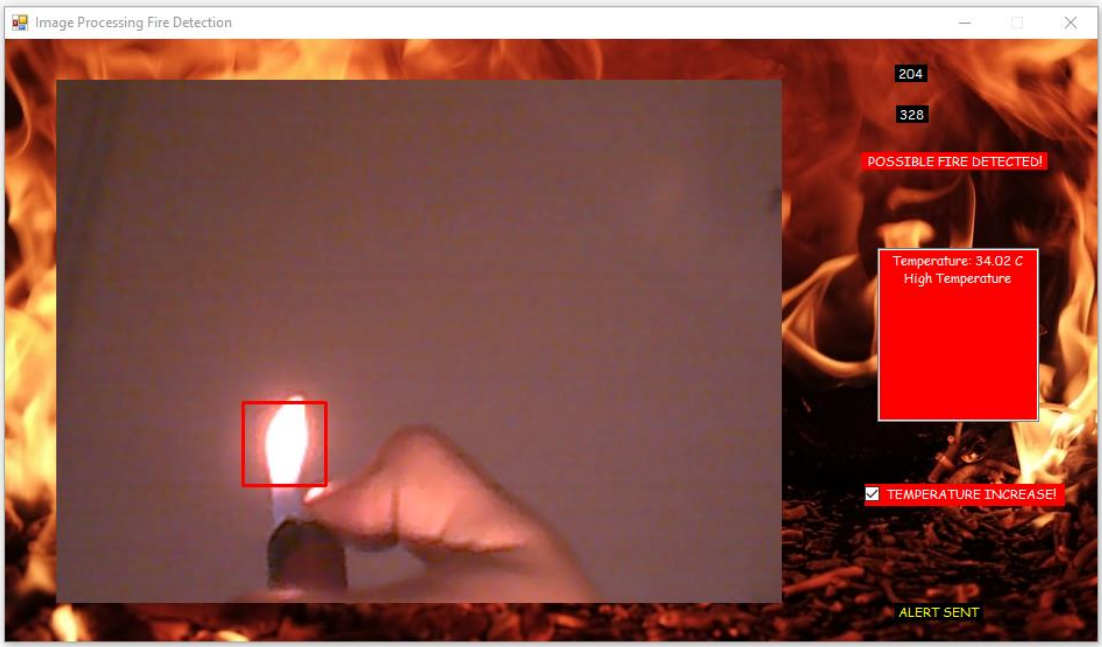
Zatim smo ispred kamere postavili plamen s upaljača kako bi simulirali vatru u prostoru. Na slici 9.2 možemo vidjeti kako sustav reagira na tu pojavu, program je otkrio plamen u signalu s kamere (područje detektiranog objekta je označeno crvenim kvadratom) te je oznaka Label4 poprimila novo stanje. Prikazuje poruku „POSSIBLE FIRE DETECTED!“ u crvenoj pozadini što korisnika upozorava da je u prostoru moguć nastanak požara. Iako je vatra opažena sustav ne šalje e-mail upozorenje jer je ispunjen samo jedan dio uvjeta za uzbunjivanje.

Kako bi provjerili i drugi uvjet, ukloniti ćemo plamen ispred kamere te zagrijati temperaturni senzor na temperaturu iznad 31.00 °C. Sustav sada prikazuje poruku u tekstualnom okviru „High Temperature“ uz iznos temperature od 32.49 °C u crvenoj pozadini uz promjenu stanja CheckBox1 u stanje *true* također uz poruku upozorenja “TEMPERATURE INCREASE!“ u crvenoj pozadini. Kao i u prošlom slučaju, e-mail uzbunjivanje nije izvršeno jer samo jedan od dva potrebna uvjeta je ispunjen.



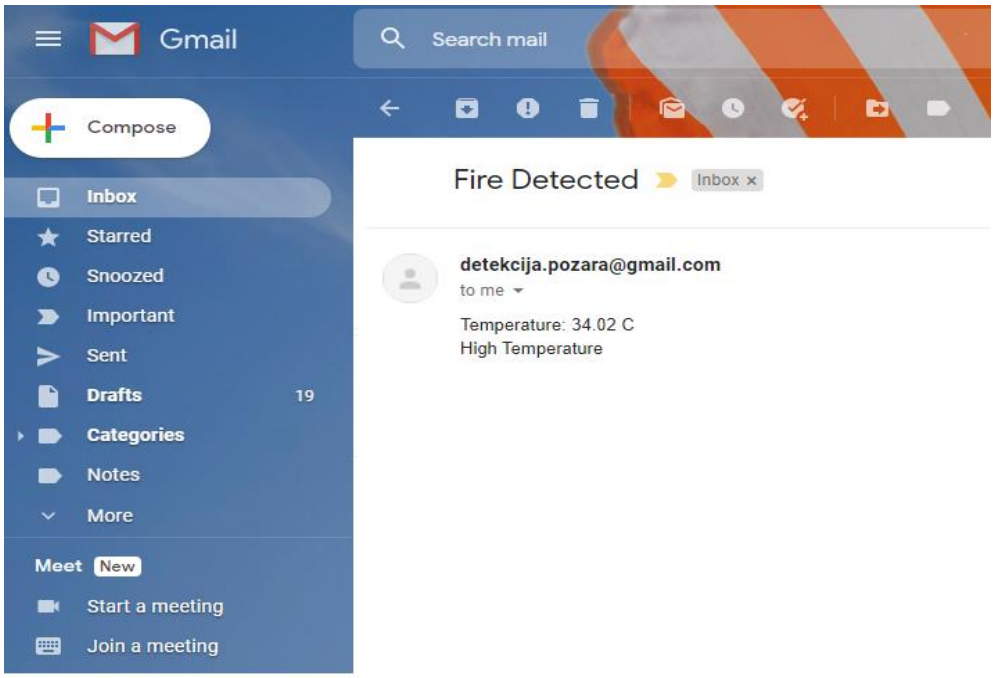
Slika 9.3 Detektiran porast temperature u prostoru

Konačno, kada prikažemo plamen na slici te uz to i zagrijemo temperaturni senzor iznad temperaturne granice sustav će poslati upozorenje korisniku.



Slika 9.4 Uzbunjivanje korisnika

Na slici 9.4 vidimo da sustav prikazuje detektiran požarni objekt na slici i porast temperature u prostoru stoga nam se u donjem desnom kutu prikazuje poruka „ALERT SENT“ koja označava da je e-mail upozorenje poslano korisniku u obliku kojeg vidimo na slici 9.5. Naslov poruke glasi „Fire Detected“ a sadržaj e-maila je iznos temperature prostora, odnosno podaci sa serijskog porta u trenutku opažanja požara.



Slika 9.5 E-mail upozorenje o detektiranom požaru u štićenom prostoru

ZAKLJUČAK

Uz mnoštvo načina za detekciju požara dostupnih na današnjem tržištu u ovom radu prikazan je sustav koji je ujedno pouzdan a i cjenovno pristupačan te je ispunio sva očekivanja. Prilikom njegova testirana u stvarnim uvjetima utvrdili smo da ovaj sustav pravovremeno reagira na porast temperature u prostoru te da gotovo sigurno detektira požarni objekt na video signalu kamere. Program dakako u nekim slučajevima lažno prijavi požar na slici ali to je očekivano za ovakav način detekcije objekata. Taj problem je uspješno riješen postavljanjem temperaturnog senzora koji osigurava da se korisnik alarmira o požaru samo onda kada dođe do povećanja iznosa temperature u prostoru jer se ovaj sustav oslanja na otkrivanje požara prema dvama produktima gorenja: svjetlosti i toplini, dok većina drugih sustava opaža samo jedno od toga.

Ovakav projekt prigodan je za upotrebu u kućanstvu ali pritom treba voditi računa o mjestu postavljanja kamere i temperaturnog senzora, odnosno temperaturnoj granici pri kojoj će sustav reagirati. Ukoliko je promatrani prostor kuhinja (često mjesto nastajanja požara u kućanstvu) moramo voditi računa o tome da temperaturnu granicu postavimo nešto više od uobičajene zbog toplinskog izvora sa štednjaka, dok drugi prostori ne zahtijevaju značajniju pažnju pri postavljanju sustava. Ukoliko želimo ovaj način detekcije požara primijeniti na većim prostorima potrebno je postaviti više temperaturnih senzora, a i kamera ukoliko je potrebno, kako bi valjano pokrili cijeli štićeni prostor.

Ovaj sustav je moguće nadograditi sa Wi-Fi modulom ESP8266 koji bi omogućio korisniku praćenje temperature u realnom vremenu bilo gdje, u bilo kojem trenutku. Umjesto e-mail upozorenja moguće je implementirati bilo koji drugi način uzbunjivanja korisnika.

LITERATURA

[1] World fire statistic 2017, https://www.ctif.org/sites/default/files/ctif_report22_world_fire_statistics_2017.pdf [18.04.2020]

[2] Digital media, https://en.wikipedia.org/wiki/Digital_media [20.04.2020]

[3] Image processing, https://www.sciencedirect.com/topics/engineering/image-processing [20.04.2020]

[4] Arduino Uno, https://en.wikipedia.org/wiki/Arduino_Uno [24.04.2020]

[5] Arduino pin diagram, http://14.eu.lifeguide-ilona-thomsen.de/arduino-pin-diagram.html [24.04.2020]

[6] BMP180, https://components101.com/sensors/bmp180-atmospheric-pressure-sensor [08.06.2020]

[7] BMP180 scheme, http://arduinolearning.com/code/bmp180-barometric-pressure-sensor-example.php [08.06.2020]

[8] Genius Eye 110 camera, https://www.cnet.com/products/genius-eye-110-web-camera-series/ [29.04.2020]

[9] Computer Vision, https://www.unite.ai/what-is-computer-vision/ [30.04.2020.]

[10] Haar cascade clasificator, http://www.willberger.org/cascade-haar-explained/ [30.04.2020]

[11] Forest fire detection system using arduino and image processing vb-net, https://www.electroniclinic.com/forest-fire-detection-system-using-arduino-and-image-processing-vb-net [02.05.2020]

[12] Microsoft Visual Studio 2010, https://en.wikipedia.org/wiki/Microsoft_Visual_Studio#2010 [05.05.2020]

[13] EmguCV, http://www.emgu.com/wiki/index.php/Main_Page [05.05.2020]

[14] OpenCV, https://opencv.org/about/?ncid=txtlnkusaolp00000595 [05.05.2020]

[15] VB-Net timers, http://www.technologyuk.net/computing/software-development/computer-programming/vbnet/timers.shtml [10.05.2020]

[16] Arduino IDE, https://en.wikipedia.org/wiki/Arduino_IDE [20.05.2020]

[17] Rapid object detection using boosted cascade of simple features, https://ieeexplore.ieee.org/abstract/document/990517 [06.06.2020]