

Detektion av dålig lukt och smuts i bilinteriör

Examensarbete inom högskoleingenjörsprogrammet för elektroteknik

Kristian Mourad & Albin Vedin



CHALMERS

Institutionen för elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021

Kristian Mourad & Albin Vedin

Handledare: Fredrik Hellström, Chalmers Tekniska Högskola & Christer Karlsson,
Volvo Cars

Examinator: Erik Ström, Chalmers Tekniska Högskola

Institutionen för elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2021

Sammanfattning

I dagens bilar är förarens sinnen en stor faktor vid många uppgifter. Då utvecklingen av autonoma fordon fortsätter behöver dessa uppgifter ersättas för att bilen ska kunna ersätta förarens roll helt. Syftet med denna rapport är att undersöka om det finns metoder för att ersätta människans sinnen för att avgöra huruvida interiören i bilen är smutsig eller luktar dåligt.

I denna rapport beskrivs hur en bildklassificerare och ett system med en gassensor konstruerades. Bildklassificerarens syfte är att kunna ersätta människans uppgift att identifiera olika typer av smuts i en interiör. Systemet med en gassensor ska ersätta delen för detektion av dålig lukt i interiören. Bildklassificeraren och gassensorn utgör en prototyp för detektion av dålig lukt och smuts. Prototypen har bra möjligheter för vidareutveckling.

Bildklassificeraren använder ett djupt neuralt nätverk för att klassificera olika typer av smuts i realtid med hjälp av en webbkamera. Bildklassificeraren kan upptäcka olika typer av smuts i olika ljusförhållanden och på olika typer av material. Ett fungerande system med en gassensor hade också konstruerats. Systemet med en gassensor kan känna av lukt från både alkohol och rök och anger ett ppm-värde som skrivs på en OLED-skärm. När ppm-värdet överstiger en acceptabel gräns avges signaler via en buzzer och dioder för att indikera en dålig lukt i interiören.

Rapporten avslutas med en diskussion där vidareutveckling av de två metoderna tas upp samt etiska frågor utifrån filmning med bildklassificerare.

Baserat på denna rapporten kan vi dra slutsatsen att med hjälp av dagens teknik och med valda tekniker så kan detektion av lukt och smuts göras. Bilen är alltså inte beroende av en förare som avgör dessa aspekter.

Abstract

In today's cars, the driver's senses are a major factor in many tasks. As the development of autonomous vehicles continues, these tasks need to be replaced in order for the car to be able to completely replace the driver's role. The purpose of this report is to investigate whether there are methods to compensate for the human senses to determine whether the interior of the car is dirty or smells bad.

This report describes how an image classifier and a system with a gas sensor were constructed. The purpose of the image classifier is to be able to replace the human task of identifying different types of dirt in an interior. The system with a gas sensor will replace the part for detecting bad odors in the interior. The image classifier and gas sensor constitute a prototype for the detection of bad odor and dirt. The prototype has good opportunities for further development.

A well-functioning image classifier, which with the help of a deep neural network can classify different types of dirt in real time with the help of a webcam was constructed. The image classifier can detect different types of dirt in different lighting conditions and on different types of materials. A working system with a gas sensor had also been designed. The system with a gas sensor can detect odors from both alcohol and smoke and indicates a ppm value written on an OLED display. When the ppm value exceeds an acceptable limit, a buzzer and diodes emit a signal indicating a bad odor in the interior.

The report concludes with a discussion where further development of the two methods is addressed as well as ethical issues based on filming with image classifiers.

Based on this report, we can conclude that with the help of today's technology and with selected techniques, detection of odor and dirt can be done. The car is thus not dependent on a driver who decides these aspects.

Förord

Den här rapporten är ett kandidatarbete på 15 högskolepoäng som är ett sista projekt inom Chalmers högskoleingenjörsprogram mekatronik. Arbetet är utfört på distans tillsammans med Volvo Cars.

Vi vill först och främst tacka Volvo Cars för den möjligheten att få vara med och arbeta tillsammans med dom. Ett extra stort tack till Christer Karlsson som har varit med och handlett oss med goda ideer och inspirerat oss fram till våra slutliga lösningar.

Vi vill också ge ett stort tack till Fredrik Hellström vid institutionen för elektroteknik som har varit med och handlett oss med rapportskrivning och kommit med tips och hjälpmedel för att vi på bästa sätt ska kunna få fram en rapport så bra som möjligt. Vi vill även passa på att tacka Sakib Sisteck för möjligheten att använda oss av laborationssalen på Chalmers lindholmen.

Kristian Mourad & Albin Vedin, Göteborg, Maj 2021

Innehåll

1	Introduktion	1
1.1	Bakgrund	1
1.2	Mål	1
1.3	Syfte	2
1.4	Frågeställning	2
1.5	Avgränsningar	2
2	Teori	3
2.1	Gassensor	3
2.1.1	Konstruktionen av en metalloxidbaserad gassensor	3
2.1.1.1	Gasavkänningslager	4
2.1.1.2	Värmepole	4
2.1.1.3	Elektroledning	4
2.1.1.4	Elektrod	5
2.1.1.5	Rörformig keramik	5
2.1.1.6	Material över avkänningsmaterialet	5
2.1.2	Hur en gassensor funkar	5
2.1.3	Mätning av gaser	6
2.2	Arduino	6
2.2.1	Arduino Uno	7
2.2.2	Arduino Unos ben	7
2.3	Beskrivning av elektroniska komponenter	8
2.3.1	Kopplingsplatta	8
2.3.2	Summer/Buzzer	9
2.3.3	I2C OLED Display	9
2.4	Bildklassificerare	11
2.4.1	Djupinlärning (Deep learning)	11
2.4.2	AlexNet	12
2.4.3	Överförbar läring (Transfer learning)	12
3	Konstruktion och genomförande	13
3.1	Påbyggnad av system med gassensor	13
3.1.1	MQ2-sensor	14
3.1.2	Detektion av en specifik gas	14
3.1.3	Buzzer och dioder	16
3.1.4	Hela kopplingen med inkopplad OLED-skärm	17

3.2	Påbyggnad av bildklassificerare	18
4	Resultat	21
4.1	Mätning av system med MQ2-sensorn vid utsättning av rök och öl . .	21
4.2	Resultat när bildklassificeraren identifierar olika typer av smuts . . .	23
5	Diskussion	27
5.1	Detektion av lukt	27
5.2	Detektion av smuts	28
5.3	Etik	28
6	Slutsats	29
	Bibliography	31
A	Appendix 1	I
A.1	Kod för system med gassensor	II

1

Introduktion

Detta kapitel presenterar bakgrunden för att besvara följande fråga: är det autonoma fordonet i tillräckligt bra skick att överlämnas till nästa användare? Det kommer också att presentera syftet med rapporten, och projektets mål kommer att delas in i flera segment som kommer att besvaras genom hela rapporten. Slutligen avslutas kapitlet med en kort introduktion av de metoder som kommer att användas under projektet.

1.1 Bakgrund

Globala biltrender är t.ex. autonoma, elektrifierade, anslutna och delade fordon. Dessa trender innefattar att flytta från ägande av fordon till transport som en tjänst vilket i sin tur kommer att påverka affärsmodeller och tekniska lösningar. Ett av Volvos mål är att främst producera helelektriska bilar. Volvo strävar redan efter att hälften av bilarna av deras försäljning kommer att vara helt elektriska i mitten av nästa decennium. Ett annat mål är att producera autonoma bilar, samtidigt som de strävar efter att en tredjedel av all försäljning ska vara autonom.

I fordon idag fyller förare och passagerare viktiga uppgifter där de använder sina sinnen för att bedöma fordonskomponenterna genom t.ex. ljud, vibrationer, lukt och syn. Få av dessa egenskaper mäts idag av sensorerna i ett fordon och detta behövs inte eftersom en förare alltid är närvarande.

I nya affärsmodeller äger användaren kanske inte fordonet, och användaren förväntas inte rapportera fordonets status. Volvo har redan börjat utveckla helt autonoma körfärdiga fordon tillsammans med det amerikanska företaget Uber. I dessa scenarier när människan inte förväntas dela information från sinnen eller vidta åtgärder, blir vissa enkla uppgifter svåra, t.ex. att avgöra när interiören är ren eller smutsig eller luktar gott eller dåligt [1].

1.2 Mål

Målet är att bestämma sensorer, tekniska lösningar och metoder för att svara på, för en människa enkel fråga: är interiören smutsig? Luktar det bra?

1.3 Syfte

I detta projekt är syftet att undersöka hur de mänskliga uppgifterna för att bedöma statusen på ett fordon's interiör kan göras av ett system som består av sensorer som mäter specifika egenskaper och ansluter dessa sensoravläsningar till människans uppfattning om rent eller smutsigt, luktar bra eller dåligt.

1.4 Frågeställning

- Är det möjligt för sensorer att kunna känna av dålig lukt i en bil?
- Är det möjligt för sensorer att kunna känna av smuts på interiören?
- Vilken typ av sensorer kan användas för att upptäcka dålig lukt och smuts i interiören?

1.5 Avgränsningar

Under projektet fokuserar vi främst på att implementera sensorerna för att förstå ett sätt att upptäcka lukt och smuts, vi kommer inte att undersöka hur man rengör inredningen. Sensorerna kommer inte att implementeras i en riktig bil, så sensorerna blir mer en prototyp som ingenjörerna som arbetar på Volvo kan fortsätta att forska på. En av de största begränsningarna är tiden: rätt sensorer att använda måste först bestämmas och sedan beställas via Volvo, vilket beräknas ta två till fem veckor.

På grund av vår begränsade kunskap när det gäller programmering av sensorer så kommer sättet de mäter lukt och smuts på vara grundläggande.

Projektet kommer att utföras under den pågående pandemin, och det mesta av arbetet kommer att göras på distans. Detta betraktas som en begränsning eftersom vi inte kommer att vara på fabriken och kunna testa sensorerna och kunna prata med andra på Volvo utom vår handledare Christer Karlsson.

2

Teori

2.1 Gassensor

En gassensor är en anordning som detekterar närvaron eller koncentrationen av gaser i atmosfären. Generellt sett så fungerar gassensorer genom att mäta koncentrationen av gasen i ett område och producera motsvarande potentialdifferens genom att ändra resistansen hos de invändiga materialen i sensorn, vilket sedan kan mätas som utspänning. De olika typerna av gaser och deras koncentrationer kan sedan uppskattas baserat på värdet på spänningen. Vilka gaser en gassensor kan upptäcka beror på avkänningsmaterialet på insidan av sensorn. Generellt kommer sensorerna som moduler med komparatorer som kan ställas in för önskat tröskelvärde för gaskoncentration. Den digitala pinnen i komparatorn visar högt värde när koncentrationen av gasen överskrider den önskade tröskeln, och då kan den analoga pinnen användas för mätning av koncentrationen av de olika gastyperna [2].

Det finns olika typer av gassensorer som används. De klassificeras som olika typer beroende på vilket avkänningsmaterial de använder. De mest använda typerna av gassensorer är [2]:

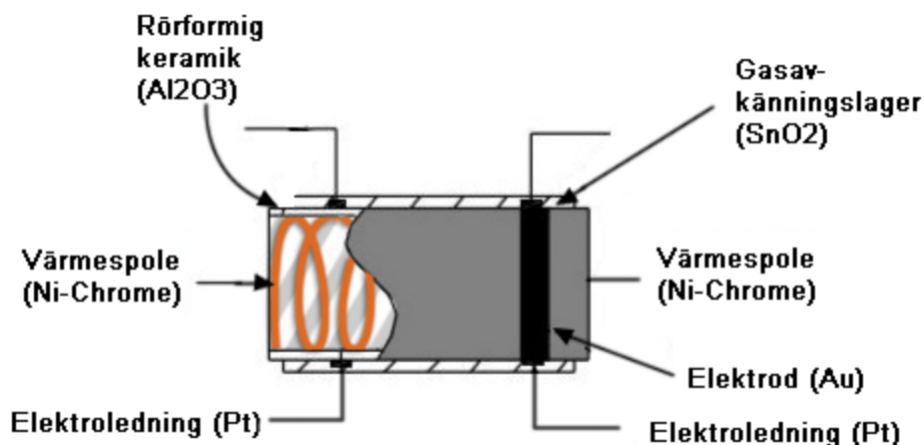
- Metalloxidbaserad gassensor.
- Optisk gassensor
- Elektrokemisk gassensor
- Kalorimetrisk gassensor
- Akustisk gassensor

Den mest använda sensorn av de som är nämnda ovan är den metalloxidbaserade gassensorn (MOS) som är den engelska översättningen för Metal oxide semiconductor. Det är en halvledare av metalloxid. Gassensorn som kommer att användas under projektets gång är en metalloxidbaserad halvledare som kallas MQ2-gassensor [3].

2.1.1 Konstruktionen av en metalloxidbaserad gassensor

Uppbyggnaden av avkänningsmaterialet i en metalloxidbaserad halvledare består av fem olika delar, vilket illustreras i figur 2.1 [3]:

1. Gasavkänningslager
2. Värmelepole
3. Elektrodledning
4. Elektrod
5. Rörformig keramik



Källa: [2]

Figur 2.1: Figur som illustrerar uppbyggnaden av en metalloxidbaserad gassensor.

2.1.1.1 Gasavkänningslager

Gasavkänningslagret är huvudkomponenten i sensorn, vilken används för att upptäcka olika förändringar i koncentrationen av gas och förändrar det elektriska motståndet. Gasavkänningslagret fungerar som en kemisk resistor. En kemisk resistor ändrar motståndets värde enligt koncentrationen av en specifik gas i miljön. När motståndet ändras ändras också strömmen. Denna komponent består av tenndioxid (SnO_2) som vanligtvis har ett överskott av elektroner. På grund av överskottet av elektroner så kommer motståndet att förändras när en giftig gas detekteras och strömmen som strömmar igenom ändras också, vilket representerar en förändring i gaskoncentrationer [2].

2.1.1.2 Värmelepole

Värmelepolens syfte är att bränna in avkänningsmaterialet, vilket ökar känsligheten och effektiviteten. Den är gjord av nickel-krom som har hög smältpunkt och det gör att den tål höga temperaturer innan den börjar smälta [2].

2.1.1.3 Elektrodledning

Eftersom avkänningsmaterialet alstrar en mycket liten ström när gasen detekteras, så är det mer kritiskt att bevara effektiviteten för att leda dessa små strömmar. För att lösa detta så har platinatrådar implementerats, vilket bidrar till att effektivisera elektronernas rörelse [2].

2.1.1.4 Elektrod

Avkänningslagrets utgång är ansluten till elektrodledningen vid denna korsning, för att utgångsströmmen ska nå rätt kurs. Elektroden är gjord av guld (Au) för att det är en väldigt bra ledare [2].

2.1.1.5 Rörformig keramik

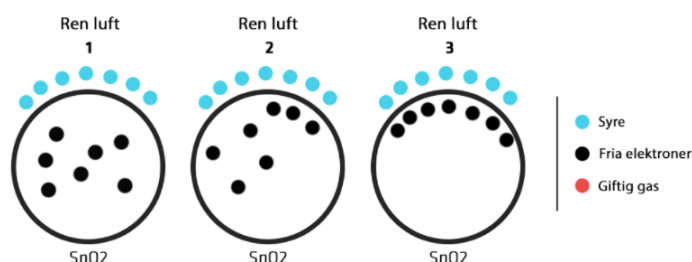
Mellan värmespolen och gasavkänningslagret finns den rörformade keramiken som är gjord av aluminiumoxid (Al_2O_3). På grund av att den har hög smältpunkt så hjälper det till att bibehålla inbränningen av avkänningslagret vilket ger en högre känslighet för gasavkänningslagret för att få en mer effektiv utgående ström [2].

2.1.1.6 Material över avkänningsmaterialet

Ett metallnät gjort av rostfritt stål används ovanpå för att skydda avkänningskomponenterna och konfigurationen. Metallnätet förhindrar damm och andra skadliga partiklar att tränga sig igenom nätet och skada gasavkänningsselementen. Metallnätet, vilket kallas antiexplosionsnätverk, säkerställer även en säkerhet mot explosioner då man mäter brandfarliga gaser [3].

2.1.2 Hur en gassensor funkar

Förmågan gassensorn har att detektera gaser är beroende av den kemiska resistorns förmåga att leda ström. Tenndioxid (SnO_2), en halvledare med fria elektroner, vilka även kallas givarelektroner, är den mest använda kemiska resistorn. I de flesta fall har atmosfären mer syre än brännbara gaser. Syrepartiklarna drar till sig de fria elektronerna i SnO_2 , vilket får dem att skjutas upp till ytan. Utgångsströmmen blir noll då inga fria elektroner är tillgängliga. Syremolekylerna (blå färg) lockar de fria elektronerna (svart färg) upp till ytan och förhindrar att den leder ström. Detta illustreras i figur 2.2.

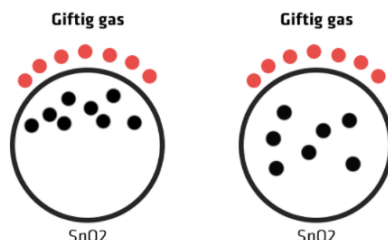


Källa: [2]

Figur 2.2: Figur som visar hur de fria elektronerna skjuts upp till ytan i tre steg. I sista steget så finns det inga fria elektroner kvar vilket medför att det inte finns någon ström.

När sensorn utsätts för giftiga eller brännbara gaser så reagerar gaserna (orange

färg) med de syrepartiklarna som finns på ytan (blå färg) och bryter den kemiska bindningen mellan syremolekylerna och de bundna elektronerna och därmed blir de fria elektroner igen. Detta medför att elektronerna återigen rör sig fritt och leder ström, vilket illustreras i figur 2.3. Storleken på strömmen blir då baserad på antalet fria elektroner. Mängden elektroner som blir fria beror på koncentrationen på den giftiga gasen, där högre koncentration ger högre utsläpp av elektroner [2].



Källa: [2]

Figur 2.3: Giftig eller brännbar gas bryter de kemiska bindningarna mellan syremolekyler och bundna elektroner, så att elektroner blir fria och leder ström igen.

MQ2-gassensorn som kommer att användas, går på 5 V DC och förbrukar cirka 800 mW. Den har ett detekteringsområde på 200 till 10000 ppm (parts per million) för gasol, rök, alkohol, propan, väte, metan och kolmonoxid [3].

2.1.3 Mätning av gaser

Termen koncentration används för att beskriva mängden gas i volym i luften vid mätning av gaser som koldioxid, syre eller metan.

Förhållandet mellan en gas och en annan mäts i ppm. Till exempel betyder 1 000 ppm kolmonoxid att av en miljon gasmolekyler är 1 000 kolmonoxid och 999 000 andra gaser [3].

2.2 Arduino

Arduino är en elektronisk plattform med öppen källkod som använder sig av en enkel hårdvara och programvara för att göra det så enkelt som möjligt att påbörja diverse projekt med olika Arduino-kort. Alla Arduino-kort fungerar på samma sätt: de läser ingångar, såsom ljus från en sensor eller en nedtryckt knapp, och konverterar dem till utgångar. Utgångarna aktiveras som svar från ingångarna. Operationerna som utförs kan vara att exempelvis när en knapp är nedtryckt ska en LED tändas eller att en motor aktiveras. Genom att skicka en serie av olika instruktioner till mikrokontrollern så kan den berätta vad Arduino-kortet skall utföra [4].

För att kommunicera med mikrokontrollern används programmet Arduino Software (IDE) som är en förenklad variant av programmeringsspråket C. När koden har

skrivits och skall överföras till Arduino-kortet så överförs koden via en USB-A till en USB-B kabel från en dator in till Arduino-kortet [4].

2.2.1 Arduino Uno

Det finns ett flertal olika Arduino-brädor att använda. Alla Arduino-brädor fungerar på ett liknande sätt, skillnaden är att olika brädor har olika antal in- och utgångar och olika mikrokontrollers. Det mest populära Arduino-kortet är Arduino Uno som visas i figur 2.4.

Arduino Uno-kortet styrs av mikrokontrollern ATmega328P. Arduino-kortet består av 6 analoga ingångar och 14 digitala in- och utgångar där 6 av dessa kan användas för PWM-styrning av motorer. Att koppla ström från Arduino-kortet till antingen en kopplingsplatta eller olika motorer är enkelt då det finns två jordade ben samt ett ben för utspänning på 5 V respektive 3.3 V [5].



Figur 2.4: Ett Arduino Uno-kort.

Arduino Uno-kortet kan drivas med en extern matning med 6 till 20 V. Om den externa spänningen till kortet är mindre än 7 V kan detta leda till att 5V-benet matar ut en spänning som är lägre än 5 V. Denna nedsatta strömförsörjning gör att kortet inte fungerar som man vill. Används en för hög spänning så kan spänningsregulatorn överhettas och skada kortet, vilket brukar förekomma vid spänningar över 12 V. Rekommendationen är att använda mellan 7 till 12 V [5].

2.2.2 Arduino Unos ben

- V_{in} : Strömförsörjning sker via detta ben om en extern strömkälla används.
- 5V: Detta ben tar emot en kontrollerad spänning på 5 V från regulatorn på kortet. Kortet kan drivas via likströmsuttaget (7-12 V), USB-kontakten (5 V) eller VIN-stiftet på kortet (7-12 V).
- 3V3: En 3.3 V utmatning som genereras av den inbyggda regulatorn.
- GND: Jordade ben.

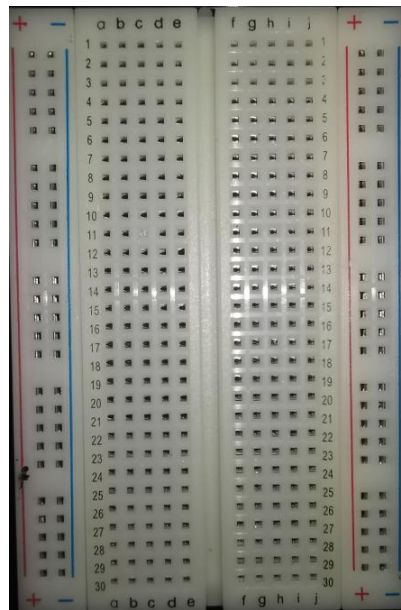
- IOREF: Spänningsreferensen som mikrokontrollern arbetar med tillhandahålls av detta ben på Arduino-kortet [5].

2.3 Beskrivning av elektroniska komponenter

2.3.1 Kopplingsplatta

En kopplingsplatta är en platta med flera små hål, vilka gör det enkelt att koppla ihop elektroniska komponenter till en elektronisk krets. Eftersom kopplingarna inte är permanenta kan de enkelt kopplas om ett misstag begås eller om en annan koppling av komponenter behöver testas. För att komponenter ska fungera att koppla till däckets behövs dom ha metallben, vilka ofta kallas för ledare. De flesta elektroniska komponenter har metallben och därmed kan de flesta kopplingar med olika komponenter testas på en kopplingsplatta [6].

Inuti kopplingsplattan finns det rader med små metallöglor där metallbenen passar in. Dessa metallöglor hålls på plats av en stödplatta. På de flesta kopplingsplattor finns det nummer, bokstäver samt tecken. Dessa markeringar hjälper till att identifiera unika hål på panelen så att den färdiga kopplingen blir enklare att förstå. De röda och blå linjerna med plus- och minustecken används ofta för att hålla koll på vilken del på däckets som är kopplat till ström och vilken som är kopplat till jord. Lägg märke till hur panelen består av uppsättningar med fem metallöglor per uppsättning. Detta betyder att varje uppsättning är ansluten mellan varandra (se figur 2.4). Exempelvis är rad 1 med kolumnerna A-E elektriskt anslutna med varandra, däremot så är rad 1 kolumnerna F-J inte anslutna till dessa kolumner. På kanterna där plus- och minustecken finns så ligger uppsättningarna på längden till skillnad från uppsättningarna i mitten. Här är däremot hela kolumn 1 med uppsättningar elektriskt ansluten till varandra till skillnad från uppsättningarna i mitten där bara en uppsättning är ansluten [6].



Figur 2.5: Kopplingsplatta som användes i projektet.

2.3.2 Summer/Buzzer

En summer är en mindre ljudanordning som ger ifrån sig ett typ av ljud när en viss spänning ansluts till modulen. Summern utgörs av ett yttre skydd med två stift där det ena stiftet kopplas till jord och det andra till strömkällan. Inuti skyddet består summern av ett piezo-element som utgörs av en rund keramisk platta. Plattan är i sig omsluten av en metall. När ström ansluts till elementet expanderar den keramiska plattan, vilket får den yttre metallen på elementet att vibrera och därmed fås ett ljud. Genom att ändra strömmen till summern ändras frekvensen, med andra ord kan ljudet vara högre eller lägre [7].

2.3.3 I2C OLED Display

Det finns många olika typer av skärmar att använda för att koppla in till Arduino. LCD (Liquid-crystal display) har varit de populäraste skärmarna att använda på grund av priset och tillgängligheten. På senare tid så har OLED (Organic Light-Emitting Diode) blivit mer önskvärda då de har: [8]

- en högre kontrast
- skarpare bild
- bred betraktningvinkel
- lägre vikt
- tunnare omfång
- stabilare material
- mindre elförbrukning

OLED-skärmarna som ofta används för Arduinoprojekt består av 128x64 respektive

128x32 separata vita OLEDs. Varje separat OLED sätts på eller stängs av via kontrollchippet SSD1306. Då skärmarna skapar sitt egna ljus så behövs inte ett bakljus, vilket bidrar till en minskad strömförbrukning för att hålla igång skärmen och en ökad kontrast. OLED-skärmarna kräver en strömkälla på 3.3 V och en 3.3 V logisk nivå för att kunna kommunicera. Strömförbrukningen varierar beroende på hur mycket av skärmen som är upplyst. I genomsnitt förbrukar skärmen cirka 20 mA från strömkällan [8]. Figur 2.6 illustrerar en OLED-skärm som används vid detta projektet.



Figur 2.6: Mini-OLED-skärm med 128x32 vita pixlar med drivrutinschippet SSD1306 och I2C-kommunikation.

OLED-skärmarna kommer oftast i moduler med ben för att kopplas till en strömkälla. SSD1306 är ett drivrutinschip som kommunicerar på olika sätt. Kommunikation via I2C-buss är den enklaste metoden att implementera. I2C använder två ledningar eller linjer, *serial clock* (SCL) och *serial data* (SDA). SCL-linjen är klocksignalen som synkroniserar den överförda datan mellan enheter på I2C-bussen och SDA-linjen bär med sig datan [10].

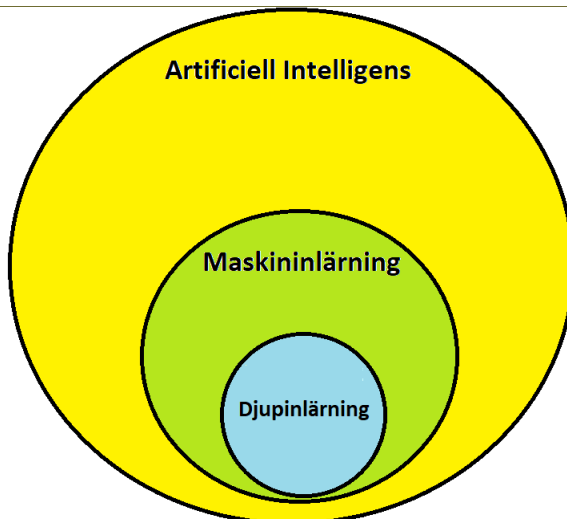
2.4 Bildklassificerare

En bildklassificerare är en produkt baserad på artificiell intelligens (AI), som utifrån en stor mängd träningsdata kan klassificera bilder i olika klasser beroende på bildens egenskaper. En bildklassificerare kan med hjälp av sannolikhetsmodeller som utvecklats av maskininlärningsstekniker göra en igenkänning och därmed en klassificering. För att träna klassificeraren behövs en stor mängd bilder för att den ska skapa sig en algoritm som på så vis känner igen egenskaperna för den specifika kategorin [11].

2.4.1 Djupinlärning (Deep learning)

Djupinlärning är en del inom maskininlärning. Maskininlärning är ett område inom det stora ämnet AI [12], detta illustreras i figur 2.7. Det speciella med maskininlärning är att med hjälp av data så kan maskininlärning producera sin egna algoritm av sig själv. Det kan beskrivas som att algoritmen utbildar sig när data matas in, och mer data ger en bättre algoritm och då ett bättre resultat [13]. Djupinlärning är som tidigare beskrivet en del inom maskininlärning. Det går att beskriva dju-

pinlärning som lager organiserade i ett neuronät. Dessa lager består utav flera neuroner. Namnet neuron kommer från människans hjärncell. Därför kan djupinlärning jämföras med hur en människa hade lagt märke till något med sina sinnen [12]. Ett djupinlärningsnätverk finns i olika varianter, vad som har visat sig vara bättre för bildklassificering är ett faltningsneuronät, eller *convolutional neural network* (CNN). Ett exempel på ett CNN är AlexNet, vilket användes för detta arbete [14].



Figur 2.7: Figur som illustrerar hur djupinlärning och maskininlärning är en del av artificiell intelligens.

2.4.2 AlexNet

AlexNet är namnet på ett redan färdigt neuronät för bildklassificering. AlexNet består utav flera lager med olika funktioner. Det första lagret är indatalagret där bilderna skickas in, sedan går bilderna igenom flera olika lager där bildbehandling sker och kategoriseringen skapas. Bilderna placeras slutligen i olika neuroner när klassificeringen är gjord [15].

2.4.3 Överförbar läring (Transfer learning)

Överförbar läring är när ett tidigare djupinlärningsnätverk används som bas för att träna en ny modell. Detta gör överförbar läring till ett smidigare sätt att starta upp sin egna modell. Hade man försökt göra ett eget nätverk från början hade det tagit betydligt längre tid. Fördelarna med att använda sig utav överförbar läring blir att amatörer kan försöka sig på sina egna modeller samt att det begärs inte lika mycket data för att få modellen stabil. I detta arbete har överförbar läring använts på nätverket AlexNet [16].

3

Konstruktion och genomförande

Här beskrivs det hur vi gick tillväga under arbetets gång. Arbetet har delats in i två delar utifrån de frågeställningar som har undersökts.

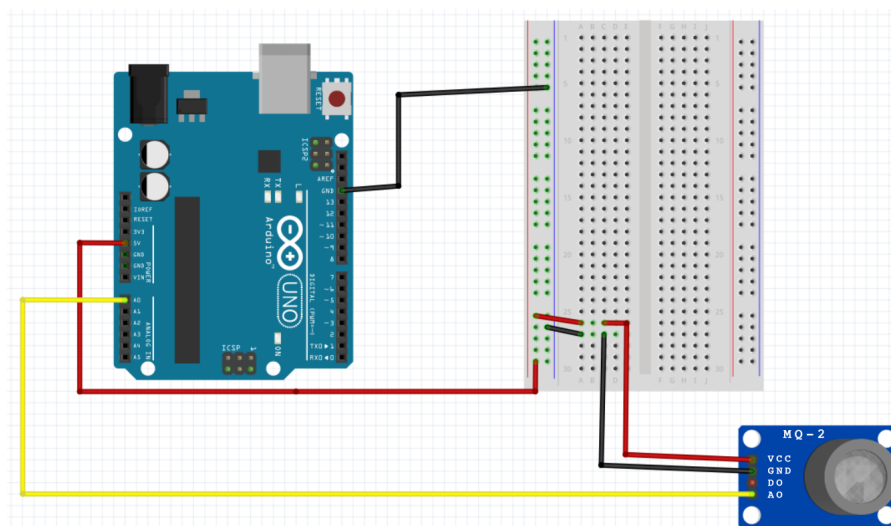
3.1 Påbyggnad av system med gassensor

Informationssökning inom ämnet detektion av lukt i interiörer på bilar påbörjades och en plan med planeringsrapport sattes upp för hur arbetet skulle gå tillväga. Efter att vi hade läst olika artiklar och studerat olika gassensorer tog vi kontakt med rekondfirmor för att ta reda på de vanligaste orsakerna till varför en bil luktar illa. Kriterierna för sensorn blev då framförallt rök och alkohollukt.

Ytterligare informationssökning gjordes och en artikel om en elektronisk näsa på Linköpings universitet blev intressant. Efter att vi hade kontakt med Linköpings Universitet för information om den elektroniska näsan blev slutsatsen att det inte fanns resurser för en elektronisk näsa då det är en väldigt hög kostnad. Till följd av detta börjades sökning göras på en billigare sensor, och efter att vi hade jämfört ett flertal olika gassensorer så gjordes beslutet att en MQ2-sensor passade bra till våra kriterier ihop med en Arduino Uno. När sammansättningen av de kopplingar som behövdes var klar kunde koden till Arduino sättas samman. För att utvärdera sensorn och få så passande mätningar som möjligt konstruerade vi en simulerad bilkupé. Mätningar på alkohol och rök kunde nu göras och därmed kan ett resultat av detektionen skrivas ner.

3.1.1 MQ2-sensor

Första steget var få igång MQ2-sensorn och påbörja kodskrivningen. Figur 3.1 visar hur den specifika kopplingen gjordes. Arduino-kortet kopplades ihop med en kopplingsplatta och kablar från 5 V och jord anslöts till kopplingsplattan för att ge en strömförsörjning till MQ2-sensorn. Därefter kopplades MQ2-sensorn till den analoga ingången A0 på Arduino-kortet.



Figur 3.1: Koppling mellan sensorn och Arduino.

Arduino-kortet kopplades upp till en strömkälla, en stationär dator i detta fallet, för att kunna läsa av de utförda mätningarna i programmet Arduino IDE. MQ2-sensorn kalibrerades för känslighet via ett kugghjul på undersidan.

3.1.2 Detektion av en specifik gas

Till en början så skrevs en kod som endast mäter spänningen från A/D-omvandlaren och skriver ut strömmen i mA. Nackdelen var att man inte kunde detektera vilken specifik gas som MQ2-sensorn utsätts för. För att kunna mäta de specifika gaserna i ppm behövde vi veta hur sensorn skiljer på olika gaser. Ur databladet för MQ2-sensorn så kunde vi se att varje specifik gas har en egen kurva med värden i en logaritmisk skala [17].

Ur figur 3.2 så valdes två punkter på varje kurva för att få lutningen på de två specifika gaserna, alkohol och rök i grafen. Två punkter valdes ut, ena vid 200 ppm och den andra vid 10000 ppm.

Följande beräkningar gjordes för att kunna beräkna ut ett ppm-värde för alkohol.

$$\begin{aligned}\log(200) &= 2.3 = x_0 && (200 \text{ är första punkten på ppm-axeln}) \\ \log(2.8) &= 0.44 = y_0 && (2.8 \text{ är första punkten på } R_s/R_0\text{-axeln}) \\ \log(10000) &= 4 = x && (10000 \text{ är sista punkten på ppm-axeln}) \\ \log(0.66) &= -0.18 = y && (0.66 \text{ är sista punkten på } R_s/R_0\text{-axeln})\end{aligned}$$

När dessa värden är kända kunde lutningen, m räknas ut:

$$m_{\text{alkohol}} = \frac{\log(y/y_0)}{\log(x/x_0)} = -0.37.$$

Lutningen för rök-kurvan beräknades på samma sätt:

$$m_{\text{rök}} = \frac{\log(-0.22/0.53)}{\log(4/2.3)} = -0.45$$

R_s/R_0 -axeln representerar koncentrationen av gas i ppm. R_0 är sensorns resistans i ren luft och R_s är sensorresistansen vid olika koncentrationer av gaser. Enligt databladet och figuren 3.2 visar ren luft att R_0 ska vara 9.83 k Ω .

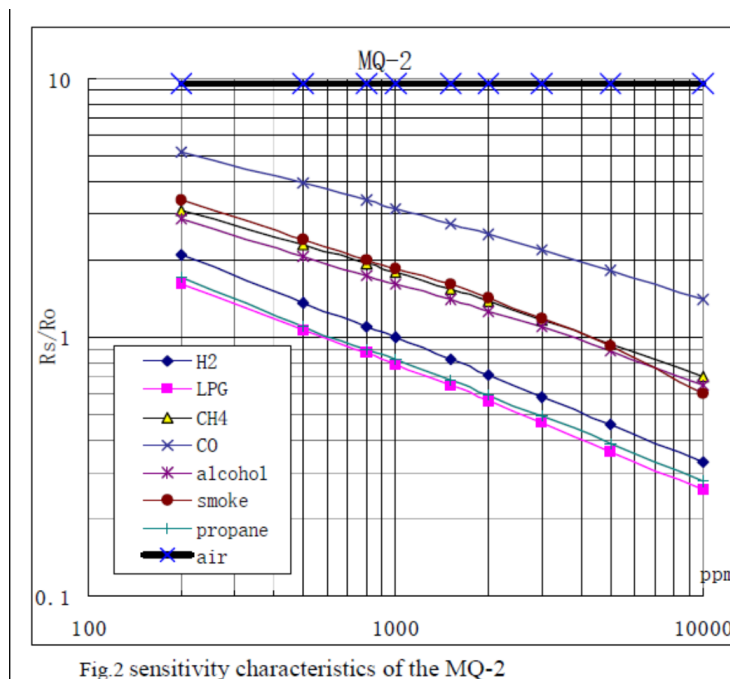


Fig.3 is shows the typical sensitivity characteristics of the MQ-2 for several gases. in their: Temp: 20 $^{\circ}$ C, Humidity: 65%, O₂ concentration 21% RL=5k Ω
 R_0 : sensor resistance at 1000ppm of H₂ in the clean air.
 R_s : sensor resistance at various concentrations of gases.

Källa: [17]

Figur 3.2: Känslighetsegenskaper för MQ2-sensorn.

Eftersom alla värden på punkterna som har beräknats är i en 10-logaritmisk skala så används funktionen nedan för att beräkna ppm-värdet. $pcurve$ används i koden för sensorn.

$$\text{Gaskoncentrationen} = (\text{pow}(10, (\log \frac{R_s}{R_o}) - \text{pcurve}[1]) / \text{pcurve}[2] + \text{pcurve}[0]) \text{ ppm}$$

där:

$$\text{pcurve}[0] = x_0$$

$$\text{pcurve}[1] = y_0$$

$$\text{pcurve}[2] = m$$

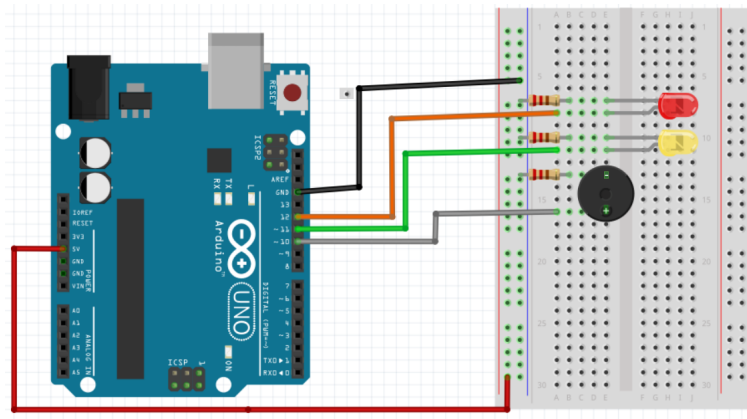
3.1.3 Buzzer och dioder

När MQ2-sensorn fungerade bra med programvaran så bestämdes det att systemet skulle kunna vara portabelt. Strömkällan ersattes nu från en stationär dator till en powerbank på 4000 mAh (milliamp Hour) som är placerad i lådan. Ett problem som uppstod var att man inte kunde läsa av diagrammet och läsa av när MQ2-sensorn upptäckte gaser. Därför bestämdes det för att införa två st dioder och en buzzer för att enkelt kunna visualisera när sensorn har upptäckt en kritisk nivå av en typ av gas. I figur 3.3 ser vi koden för dioderna och buzzern.

```
if(MQ2Procent2(MQ2Rs(MQ2)/Ro,Alkohol)>500){
    digitalWrite(redLampa, HIGH);
    digitalWrite(gulLampa, LOW);
    tone(buzzer, 1000, 200);
}
else
{
    digitalWrite(redLampa, LOW);
    digitalWrite(gulLampa, HIGH);
    noTone(buzzer);
}
```

Figur 3.3: Kod som beskriver när den röda dioden samt buzzern aktiveras.

Figur 3.4 visualiserar kopplingen mellan Arduino, dioderna och buzzern. En röd och en gul diod kopplades in till kopplingsplattan med två st 220 k Ω resistorer till Arduino-kortet som digitala utsignaler. Den gula dioden lyser konstant så länge gaspartiklarna i luften inte överstiger det kritiska värdet. Den röda lampan lyser när ppm-värdet överstiger värdet som sattes upp på 500 ppm i koden. En buzzer kopplades in för att kunna höra någon form av larm när systemet detekterar en gas. Buzzern och den röda dioden aktiveras samtidigt sensorn har upptäckt en kritisk nivå av en gas, alltså när ppm-värdet överstiger det önskade värdet.



Figur 3.4: Kopplingar till diod och buzzer.

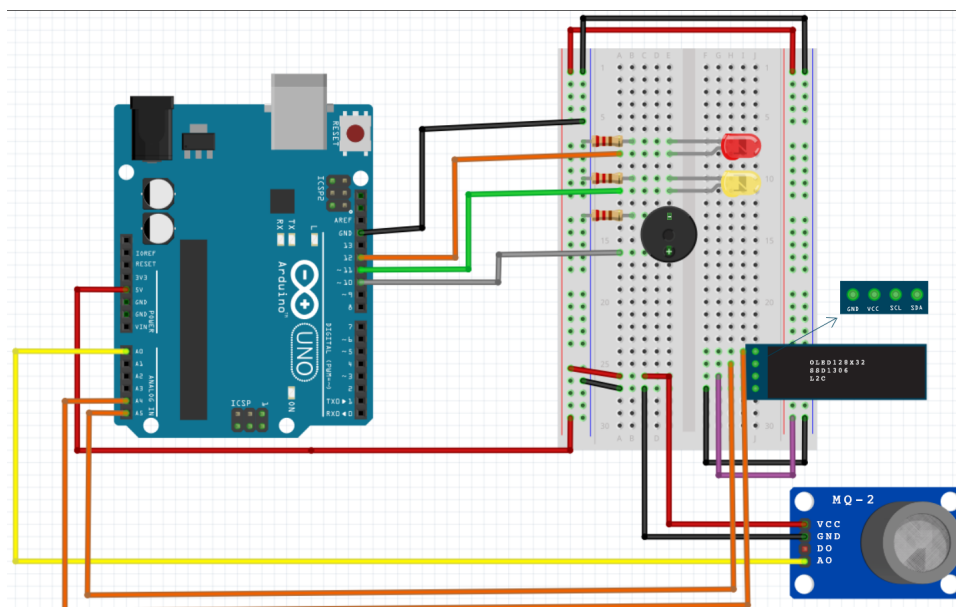
3.1.4 Hela kopplingen med inkopplad OLED-skärm

När systemet var portabelt så upptäcktes ganska snabbt ett problem med det. Man kunde endast se när gaspartiklarna i luften översteg det önskade värdet genom att buzzern larmade samt att den röda dioden började lysa. På grund av detta kopplades en display in så att man ständigt kan läsa av ppm-värdet av gaskoncentrationen i luften. I figur 3.5 ser vi hela kopplingen för gassensorn.

En OLED-skärm med I2C-koppling kopplades in till kopplingsplattan och Arduino. Den kopplades till de analoga ingångarna A4 samt A5 på Arduino-kortet, jord och 5 V kopplades in för strömförsörjning.

När displayen var inkopplad så laddades kod upp. Adafruit, ett bibliotek som redan fanns inne i programmet Arduino IDE, laddades ned till Arduino. När biblioteket var nedladdat underlättades programmeringen av displayen och istället för att få värden på datorskärmen kunde vi nu få ppm-värden på displayen.

För att kunna göra mätningarna i en miljö som liknar en bilkupé användes en låda i plast med måtten $26 \times 14 \times 15$ cm, där sensorn tillsammans med resterande kopplade komponenter placerades. Detta gjordes för att kunna utföra tester med stängt lock för att simulera en enkel bilkupé.



Figur 3.5: Hela koppling med alla komponenter.

3.2 Påbyggnad av bildklassificerare

När andra delen av arbetet hade studerats påbörjades informationssökning inom ämnet av smuts i interiören i bilar. Här behövdes information om vad som klassas som smuts i interiören. Intervjuer med privatpersoner och taxichaufförer hölls och då kunde det avgöras vad som skulle detekteras. Nästa steg var hur det skulle detekteras, vi studerade olika metoder på detta och kom fram till att använda oss av ett enkelt sätt att göra sin egna klassificerare med överförbar läring. AlexNet blev nätverket som vi använde oss av och tillslut hade vi en klassificerare som kunde detektera de kategorier av smuts som vi hade satt upp.

Konstruktionen av klassificeraren som användes under detta projektet var en anpassning av Joe Hicklins klassificerare. Hicklins är en programvarutvecklare på Mathworks [18].

Klassificeraren anpassades efter kriterierna som sattes upp under projektets gång och började tränas därefter. En mapp som innehöll fem undermappar skapades. De fem undermapparna skapades för att träna klassificeraren att detektera de fem olika klasserna: smulor, godispapper, fläckar, rent och läskburkar. Varje undermapp innehöll mellan 600-1500 bilder på den specifika klassen. För att få en bra träningsdata så togs bilder både via telefonkamera och online. Klassen med bilder på vad som kallas rent användes för att kunna ha någon form av referens så att bildklassificeraren ständigt visar rent sålänge inte en annan klass upptäcks. För att kunna identifiera om klassificeraren verkligen kunde upptäcka klasserna och klassificera dem korrekt så användes det olika bilder under träningen jämfört med när den var kopplad till kameran. Vi ville att klassificeraren skulle kunna känna igen de olika kategorierna

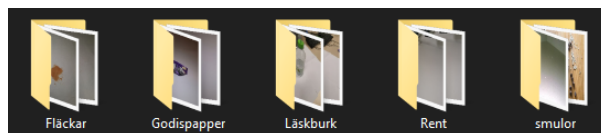
utan att ha tränat på en specifik bild. Träningsdatan för läskburkar innehöll inga bilder på blåa eller röda burkar för att sedan kunna använda burkar med de färgerna under testdatan.

För att programmet skall hitta mappen med alla undermappar användes funktionen "ImageDataStore", vilken gör att koden går in i den specifika mappen och läser av alla undermappar och delar upp dem i klasser med en titel, där titeln är baserad på undermappens namn. Detta visas i figur 3.6 och 3.7.

```
%% Lägga upp träningsdatan. "Datset1 är en mapp med alla subfolders i.  
%varje subfolder ha ca 800 bilder var på en specifik klass"  
allImages = imageDatastore('Datset1', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');  
[trainingImages, testImages] = splitEachLabel(allImages, 0.8, 'randomize');
```

Figur 3.6: Funktionen ImageDataStore.

3. Konstruktion och genomförande



Figur 3.7: Alla undermappar. Varje undermapp innehåller 600-1500 bilder.

AlexNet består av 25 lager där varje lager har en specifik funktion som gör så att AlexNet kan identifiera olika objekt. Det 23e lagret består av 1000 olika neuroner, där varje neuron är en kategori. Detta lagret anpassas till våra kriterier och ändras till endast fyra. Det 25e lagret är tränat på AlexNets egna nätverk, det ändras också för att kunna skriva ut våra egna resultat. Figur 3.8 visar hur klassificeraren tränas för att lära sig en specifik kategori.

```
Training on single CPU.
Initializing input data normalization.
=====
| Epoch | Iteration | Time Elapsed | Mini-batch | Mini-batch | Base Learning |
|       |          | (hh:mm:ss)  | Accuracy   | Loss        | Rate          |
|=====|=====|=====|=====|=====|=====|
| 1 | 1 | 00:00:02 | 26.56% | 2.8578 | 0.0010 |
| 2 | 50 | 00:01:19 | 100.00% | 0.0030 | 0.0010 |
| 4 | 100 | 00:02:38 | 100.00% | 0.0002 | 0.0010 |
| 5 | 150 | 00:03:57 | 100.00% | 0.0004 | 0.0010 |
| 7 | 200 | 00:05:19 | 100.00% | 4.2181e-05 | 0.0010 |
| 8 | 250 | 00:06:40 | 100.00% | 0.0008 | 0.0010 |
| 10 | 300 | 00:07:59 | 100.00% | 6.1724e-05 | 0.0010 |
| 11 | 350 | 00:09:18 | 100.00% | 0.0012 | 0.0010 |
| 13 | 400 | 00:10:38 | 100.00% | 0.0022 | 0.0010 |
| 15 | 450 | 00:11:58 | 100.00% | 0.0017 | 0.0010 |
| 16 | 500 | 00:13:17 | 100.00% | 0.0001 | 0.0010 |
| 18 | 550 | 00:14:35 | 100.00% | 8.0113e-05 | 0.0010 |
| 19 | 600 | 00:15:55 | 100.00% | 0.0002 | 0.0010 |
| 20 | 640 | 00:16:58 | 100.00% | 2.1908e-05 | 0.0010 |
|=====|=====|=====|=====|=====|=====|

accuracy =
0.9962
```

Figur 3.8: Bildklassificeraren tränas och kör 20 cykler (epochs) för att få så bra precision som möjligt.

Slutligen kopplas en webkamera in så att klassificeraren kan ta den tränade indatan och användas i realtid. Webkameran anslöts och en add-on lades in i Matlab som stödjer uppkoppling av webkameror. När programmet körs så ser man att med hjälp av webkameran så visar den upp en titel med de olika klasserna när en klass filmas och på så sätt identifierar de olika smutsklasserna.

4

Resultat

I detta kapitel redovisas alla mätningar och resultat från systemet med en gassensor samt bildklassificeraren.

4.1 Mätning av system med MQ2-sensorn vid utsättning av rök och öl

Öl hälldes i en plastkopp som vi förde ner och placerade i samma låda som MQ2-sensorn låg i. När MQ2-sensorn kände av dessa gaser så kunde vi se en ökning av ppm av både alkohol och rök, se figur 4.1.

```
20:40:49.305 -> Alkohol:375ppm
20:40:49.586 -> Smoke:503ppm
20:40:50.763 ->
20:40:50.998 ->
20:40:51.187 -> Alkohol:208ppm
20:40:51.423 -> Smoke:334ppm
20:40:52.606 ->
20:40:52.844 ->
20:40:53.032 -> Alkohol:1098ppm
20:40:53.270 -> Smoke:1550ppm
20:40:54.215 ->
20:40:54.453 ->
20:40:54.641 -> Alkohol:1478ppm
20:40:54.879 -> Smoke:1647ppm
20:40:55.822 ->
20:40:56.057 ->
20:40:56.245 -> Alkohol:1721ppm
20:40:56.483 -> Smoke:1952ppm
20:40:57.426 ->
20:40:57.664 ->
20:40:57.805 -> Alkohol:1937ppm
20:40:58.090 -> Smoke:2016ppm
20:40:59.027 ->
20:40:59.261 ->
20:40:59.450 -> Alkohol:1326ppm
20:40:59.684 -> Smoke:1367ppm
20:41:00.624 ->
20:41:00.858 ->
20:41:01.047 -> Alkohol:1339ppm
20:41:01.278 -> Smoke:1557ppm
20:41:02.217 ->
20:41:02.453 ->
20:41:02.641 -> Alkohol:1268ppm
20:41:02.877 -> Smoke:1495ppm
20:41:03.820 ->
20:41:04.057 ->
20:41:04.246 -> Alkohol:2126ppm
20:41:04.482 -> Smoke:2091ppm
```

Figur 4.1: Mätningar när MQ2-sensorn blir utsatt för alkohol.

4. Resultat

För att göra mätningar på rök tändes en cigarett och som sedan lades i ett askfat som vi förde ner och placerade i samma låda som MQ2-sensorn. Resultatet av dess ppm-värden ser vi i figur 4.2.

```
20:51:43.433 -> Alkohol:1421ppm
20:51:43.717 -> Smoke:1574ppm
20:51:44.615 ->
20:51:44.897 ->
20:51:45.038 -> Alkohol:1438ppm
20:51:45.322 -> Smoke:1626ppm
20:51:46.219 ->
20:51:46.505 ->
20:51:46.647 -> Alkohol:1521ppm
20:51:46.930 -> Smoke:1665ppm
20:51:47.821 ->
20:51:48.105 ->
20:51:48.247 -> Alkohol:1539ppm
20:51:48.530 -> Smoke:1720ppm
20:51:49.430 ->
20:51:49.712 ->
20:51:49.854 -> Alkohol:1619ppm
20:51:50.138 -> Smoke:1720ppm
20:51:51.041 ->
20:51:51.273 ->
20:51:51.462 -> Alkohol:1702ppm
20:51:51.745 -> Smoke:1809ppm
20:51:52.644 ->
20:51:52.879 ->
20:51:53.069 -> Alkohol:1722ppm
20:51:53.349 -> Smoke:1869ppm
20:51:54.252 ->
20:51:54.489 ->
20:51:54.679 -> Alkohol:1751ppm
20:51:54.915 -> Smoke:1930ppm
20:51:55.858 ->
20:51:56.091 ->
20:51:56.280 -> Alkohol:1821ppm
20:51:56.518 -> Smoke:1948ppm
```

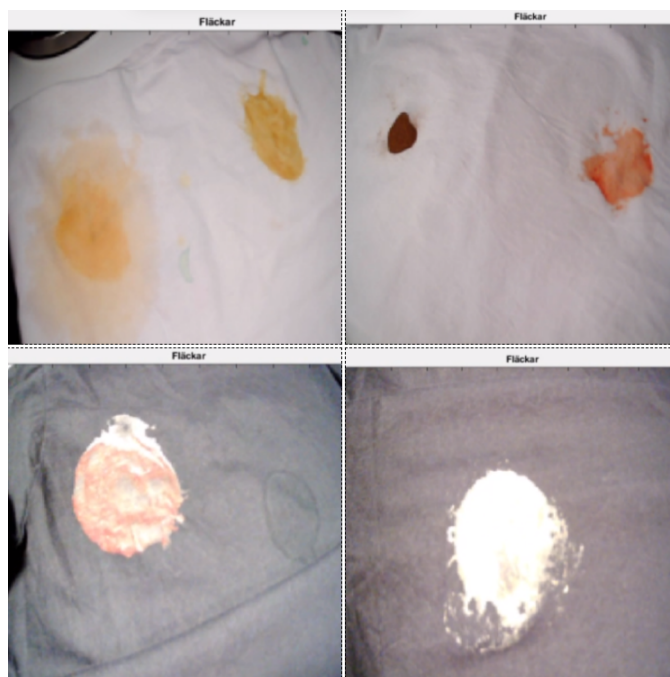
Figur 4.2: Mätningar när MQ2-sensorn blir utsatt för rök.

4.2 Resultat när bildklassificeraren identifierar olika typer av smuts

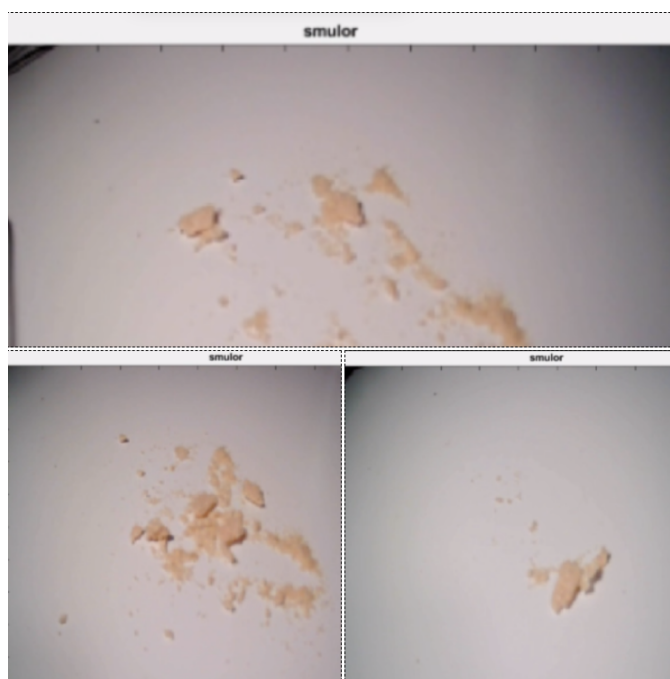
När nätverket hade tränats med alla bilder så aktiverades webbkameran och tester kunde påbörjas. Klassificeraren gav bra resultat, den lyckades identifiera alla klasser vid olika ljusförhållanden, vinklar och bakgrunder, se figurerna 4.3, 4.4, 4.5, 4.6 och 4.7. För att testa att klassificeraren verkligen förstod vad den ska upptäcka så laddades inga bilder på blåa läskburkar upp när den skulle träna. När vi testade att detektera en blåfärgad burk så lyckades den göra det. Samma metod på tester gjordes på alla fyra klasser och gav bra resultat.



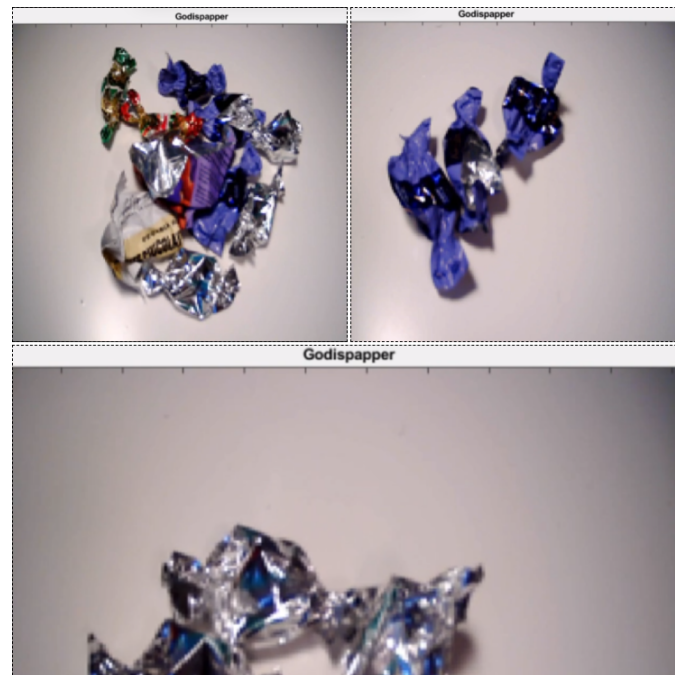
Figur 4.3: Skärmbild på detektion av en läskburk.



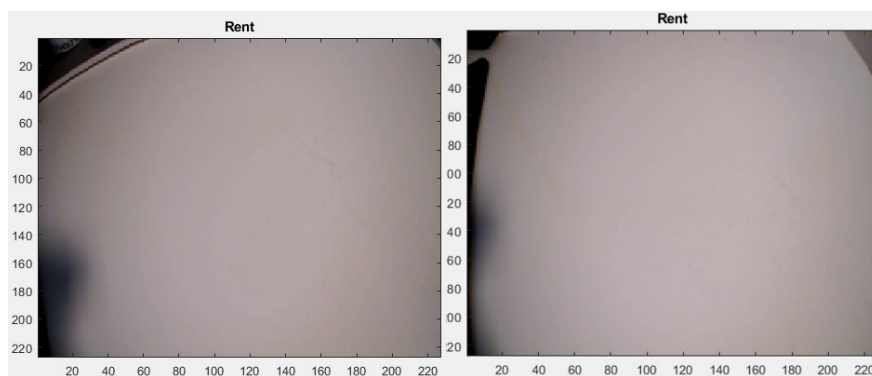
Figur 4.4: Skärmbild på detektion av olika fläckar på två olika färger av tyg som skall simulera ljus och mörk interiör.



Figur 4.5: Skärmbild på detektion av smulor.



Figur 4.6: Skärmbild på detektion av olika typer av godispapper.



Figur 4.7: Skärmbild när bildklassificeraren detekterar det som klassas som rent för att kunna ha ett underlag på vad som är rent eller smutsigt.

5

Diskussion

Målet med projektet var att undersöka om det finns metoder för olika sensorer att ersätta de mänskliga uppgifterna för bedömning av status i interiören på en bil: är bilinteriören smutsig och luktar den illa? Under projektets gång hittades lösningar till både detektion av smuts med hjälp av sensorer samt att med hjälp av sensorer kunna upptäcka dålig lukt i en interiör.

5.1 Detektion av lukt

Under projektets gång användes en gassensor för detektion av dålig lukt. Sensorn passade just de specifika kriterierna som sattes upp.

Under vårt projekt kunde vi identifiera både alkohol och rök med sensorn. Vad den inte kunde göra var att skilja på vad som faktiskt är rök och vad som faktiskt är alkohol. Detta beror på att kurvan för rök alltid är högre än kurvan för alkohol för gaskoncentrationer mindre än ca 7000ppm enligt figur 3.2. På grund av detta kommer en R_s/R_0 -mätning visa ett högre ppm-värde på rök jämfört med alkohol. Detta leder till att det är omöjligt att definitivt skilja de två gaserna åt med MQ2-sensorn och metoden som användes i detta projekt.

Det finns möjligheter för att vidareutveckla systemet för igenkänning av dåliga lukter. En elektronisk näsa som har nämnts tidigare är antagligen det bästa alternativet för det. Den elektroniska näsan kan anpassas till de specifika behoven, och kan ha ett mycket bredare användningsområde än MQ2-sensorn. Med den elektroniska näsan skulle man kunna utveckla detektionen av lukt och kunna upptäcka exempelvis spya, svettluft och mycket fler illaluktande dofter som kan förekomma i en interiör. Den elektroniska näsan anpassas med olika antal sensorer för det specifika ändamålet. Användningsområdet för den elektroniska näsan kan vara allt från att identifiera exakt vilken tidpunkt det är dags att skörda vindruvor, särskilja gris från nötkött, lukta sig till bomber, narkotika, elektroniska näsan kan även upptäcka äggstockstumörer och mycket mer [19].

5.2 Detektion av smuts

En bildklassificerare har använts för att identifiera smuts i interiören på bilar, vilket var ett av målen med rapporten. Med bildklassificeraren kunde vi identifiera de kategorierna vi hade satt upp som smuts.

För att vidareutveckla denna metoden hade det behövts fler resurser samt längre tid för arbetet. För att vidareutveckla en bildklassificerare hade jämförelse av olika neuronnät kunnat göras och på så sätt hade beslut kunnat tagits om vilket nät som hade passat bäst för just detta arbetet. Vad som mer hade varit vidareutvecklande av klassificeringen hade varit om nätverket hade tränats ännu mer och på det sättet få en högre noggrannhet, med andra ord en högre procenthalt på att identifiera de specifika objekten. Vad vi även hade velat införa är att när kameran identifierar något objekt ska den vara en viss procent säker på vad den har hittat, innan den skriver ut resultatet. Exempelvis om kameran har identifierat en typ av smuts, ska den bara säga att det är smuts om den är 80 procent säker på det.

5.3 Etik

Det finns många frågor som berör ämnet artificiell intelligens inom objekt-detektering. En fråga som borde diskuteras innan man börjar införa en form av sensor som upptäcker föremål i bilar är just att, vad tycker personer om att ständigt ha en eller flera kameror riktade mot sig när de exempelvis har beställt en taxi som skall transportera dem från en destination till en annan? Är det etiskt rätt att företag får tillgång till videofilmer eller bilder på personer som använder sig av tjänsten?

Ett argument är att företag redan använder sig av videoövervakning i butiker, men på grund av att AI är ett så nytt och okänt område för många så blir det mer fundersamt. Ett alternativ för videoövervakning kan vara att kamerorna behöver inte vara på under färden, utan det räcker att bilen kontrollerar interiören mellan resor då det inte finns en passagerare i. Enligt vårt perspektiv så är tanken att implementera AI i bilar som mäter av hur smutsig en interiör är väldigt gynnsamt. Det gynnar både företaget i fråga samt personer som använder sig av tjänsten. Fördelen med att implementera in AI som kommunicerar med de självkörande bilarna är att de själva kan avgöra när interiören är tillräckligt smutsig för att åka in till en biltvätt vid behov. Att bilarna endast åker in till en biltvätt vid behov gynnar inte bara företaget ekonomiskt, det blir även mer miljövänligt. Bilarna behöver inte åka onödiga sträckor vilket minskar elförbrukningen, slitage och laddningsintervallen av bilarna, vilket i sin tur leder till att den rena energin kan användas till annat. Det hade även gynnat personerna som åker i bilarna i form av att de alltid vet att de kommer boka en tillräckligt ren bil att färdas i.

6

Slutsats

Baserat på frågeställningen i avsnitt 1.4 så kan vi slutligen ge följande svar: att med hjälp av en gassensor så kan detektion av en specifik lukt göras. Det handlar till största del om att ha en passande sensor för den specifika situationen och konstruera därefter. Efter att ha använt oss av en MQ2-sensor vet vi att den kan upptäcka de specifika gaserna i fråga. Sensorn kan dock inte skilja och specifikt visa vad som upptäcks via metoden som vi använde.

Vi vill gärna poängtera att en elektronisk näsa i första hand är det bästa alternativet för detektion av lukt i dessa miljöer, och med vidare studier kan en sådan konstrueras.

Efter att ha studerat sensorer inom detektion av smuts kunde vi dra slutsatserna att en bildklassificerare var det bästa alternativet för detektion av smuts för att den ständigt utvecklas och kan lära sig nya kategorier samt att den bara kan bli bättre när mer data matas in.

Detta är två breda områden med många olika tillvägagångssätt, därför ser vi detta arbete som en förstudie för hur man kan med två enkla metoder undersöka lukt och smuts. Det vill säga att om mer tid hade getts för arbetet så hade flera förbättringsmöjligheter samt en bredare undersökning kunnat göras. I diskussionen (avsnitt 5) tar vi upp vidareutveckling av systemen och hur man hade kunnat gå tillväga för att utveckla dessa metoder.

Litteraturförteckning

- [1] Samuelsson, H (2019) Volvo Cars and Uber present first autonomous drive-ready production car [hämtat 2021-02-10]. Tillgänglig: <https://group.volvocars.com/news/future-mobility/2019/volvo-and-uber-present-autonomous-drive-ready-xc90>
- [2] Components101 (2019) Introduction to Gas Sensors [hämtat 2021-03-10]. Tillgänglig: <https://components101.com/articles/introduction-to-gas-sensors-types-working-and-applications>
- [3] Lastminuteengineers (2021) How MQ2 Gas/Smoke Sensor Works [hämtat 2021-03-10]. Tillgänglig: <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/>
- [4] Arduino (2021) Introduction [hämtat 2021-03-20]. Tillgänglig: <https://www.arduino.cc/en/guide/introduction>
- [5] Arduino (2021) Arduino Uno Rev3 [hämtat 2021-03-20]. Tillgänglig: <https://store.arduino.cc/arduino-uno-rev3>
- [6] Science buddies (2021) How to use a breadboard [hämtat 2021-03-19]. Tillgänglig: <https://www.sciencebuddies.org/science-fair-projects/references/how-to-use-a-breadboard#what-breadboard>
- [7] Cuidevices (2021) Piezoceramic Element Structure [hämtat 2021-03-19]. Tillgänglig: <https://www.cuidevices.com/product-spotlight/piezo-and-magnetic-buzzers#3>
- [8] Adafruit (2012) Overview [hämtat 2021-04-12]. Tillgänglig: <https://learn.adafruit.com/monochrome-oled-breakouts?view=all#power-requirements>
- [9] Cdon (2021) Mini OLED display med 128x32 vita pixlar. I2C SSD1306 [hämtad 2021-04-13]. Tillgänglig: <https://cdon.se/bygg-verktyg/mini-oled-display-med-128x32-vita-pixlar-i2c-ssd1306-p49791064>
- [10] Dejan (2021) How It Works [Hämtat 2021-04-12]. Tillgänglig: <https://howtomechatronics.com/tutorials/arduino/how-i2c-communication-works-and-how-to-use-it-with-arduino/>
- [11] Houg Tung, G (2020) Accelerated training of an image classifier [hämtat 2021-04-15]. Tillgänglig: <https://appft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetacgi%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=AND&d=PG01&s1=20200410287.PGNR.&OS=DN/20200410287&RS=DN/20200410287>
- [12] Goodfellow I, Bengio Y, Courville A (2016) Chapter 12 [Hämtat 2021-04-15]. Tillgänglig: <https://www.deeplearningbook.org/>

- [13] Domingo, P (2012) A Few Useful Things to Know About Machine Learning [hämtat 2021-04-15]. Tillgänglig: <https://homes.cs.washington.edu/~pedrod/papers/cacm12.pdf>
- [14] Nielsen, M (2019) Neural Networks and Deep Learning [hämtat 2021-04-15]. Tillgänglig: <http://neuralnetworksanddeeplearning.com/>
- [15] Krizhevsky A, Sutskever I, E.Hinton G (2012) ImageNet Classification with Deep Convolutional Neural Networks [hämtat 2021-04-15]. Tillgänglig: <https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>
- [16] Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation [hämtat 2021-04-15]. Tillgänglig: <https://arxiv.org/pdf/1311.2524.pdf>
- [17] Hanwei Electronics CO. (2021) Technical Data MQ-2 Gas Sensor [hämtat 2021-05-11]. Tillgänglig: <https://www.mouser.com/datasheet/2/321/605-00008-MQ-2-Datasheet-370464.pdf>
- [18] Hicklin, J (2017) Deep Learning with MATLAB [hämtat 2021 04-15]. Tillgänglig: <https://www.mathworks.com/videos/deep-learning-with-matlab-transfer-learning-in-10-lines-of-matlab-code-1487714.html>
- [19] Larsson, P (2010) Elektronisk näsa upptäcker cancer, bomber och vin [hämtat 2021-04-27]. Tillgänglig: <https://www.kth.se/aktuellt/nyheter/elektronisk-nasa-upptacker-cancer-bomber-och-vin-1.70940>

A

Appendix 1

I detta appendix presenteras MATLAB-koden som användes för bildklassificeraren. Här laddas AlexNets olika lager ned och ersätts sedan med det antal kategorier som vi vill ha till vår klassificerare. Sedan utförs träning för de olika kategorierna för att sedan starta webbkameran för att kunna klassificera.

```
1 % Copyright 2017 The MathWorks, Inc.
2
3 %% ladda in det intränade neuronätet
4
5 alex = alexnet;
6 layers = alex.Layers
7
8 %% Anpassa nätverket till x antal klasser, 4 i detta fall
9 layers(23) = fullyConnectedLayer(4);
10 layers(25) = classificationLayer
11
12 %% Lägga upp träningsdatan. "Datset1 är en mapp med alla subfolders i, varje subfolder ha ca 800 bilder var på en specifik klass"
13 allImages = imageDatastore('Datset1', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');
14 [trainingImages, testImages] = splitEachLabel(allImages, 0.8, 'randomize');
15
16
17
18 %% Träna nätverket igen, 20 cyklar, Epochs.
19 opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
20 myNet = trainNetwork(trainingImages, layers, opts);
21
22 %% .Mäter noggrannhet, mål är 100%
23 predictedLabels = classify(myNet, testImages);
24 accuracy = mean(predictedLabels == testImages.Labels)
25 %% Kopplad kamera som klassificerar varje ram i videofilmen och skickar ut en label
26 while true
27     picture = camera.snapshot; % Ta en bild
28     picture = imresize(picture, [227,227]); % skala om bilden till 227x227
29
30     label = classify(myNet, picture); % Klassificera bilden
31
32     image(picture); % Visa bilden
33     title(char(label)); % Visa label/namn
34     drawnow;
35 end
36
```

Figur A.1: Kod för klassificerare.

A.1 Kod för system med gassensor

Första delen av koden för gassensorn presenteras i figur A.2 där olika variabler definieras.

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define OLED_RESET -1
Adafruit_SSD1306 display(OLED_RESET);

int redLampa = 12;
int gullLampa = 11;
int buzzer = 10;
int MQ2 = 0;
int RL = 5;
int RorenLuft = 9.83; //Rs/R0 för MQ2-sensor
float Ro;

float AlkoholKurva[3] = {2.3,0.44,-0.37};
float SmokeKurva[3] = {2.3,0.53,-0.45};
#define Alkohol (0)
#define Smoke (1)

#define KalibreringsProver (25) //Antal prover under kalibreringsfasen
#define KalibreringsInterval (500) //Tid i millisekunder mellan proverna
#define AntalProver (50) //Antal prover under mätning av gas
#define ProvInterval (5) //Tid mellan varje prov av gas
```

Figur A.2

Nästa del är setup, här talar vi om för Arduino vilka komponenter som den ska arbeta med och det är även här som kalibreringen av gassensorn sker för att se att gasen befinner sig i ren luft.

```
void setup() {
  pinMode(redLampa, OUTPUT);
  pinMode(gullLampa, OUTPUT);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(1,5);
  display.print("Kalibrering");
  display.display();
  Serial.print("Kalibrering");
  Ro = MQ2Kalibrering(MQ2);
  display.clearDisplay();
  display.setCursor(1,5);
  display.print("Klart");
  display.setCursor(1,15);
  display.print("Ro=");
  display.print(Ro);
  display.print("kOhm");
  display.display();
  Serial.print("Klart");
  Serial.print("Ro=");
  Serial.print(Ro);
  Serial.print("kOhm");
  Serial.print("\n");
  delay(5000);
}
```

Figur A.3

Loop-delen är den delen som Arduino ständigt kör. Här kollar Arduino funktionerna och skriver ut de värden som gassensorn hittar.

```
void loop() {
  display.clearDisplay();
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.setCursor(1,5);
  display.print("Alkohol: ");
  display.print(MQ2Procent2(MQ2Rs(MQ2)/Ro,Alkohol));
  display.print("ppm");
  display.display();
  Serial.print("\n");
  display.setCursor(1,15);
  display.print("Smoke: ");
  display.print(MQ2Procent2(MQ2Rs(MQ2)/Ro,Smoke));
  display.print("ppm");
  Serial.print("\n");
  display.display();

  Serial.print("Alkohol:");
  Serial.print(MQ2Procent2(MQ2Rs(MQ2)/Ro,Alkohol));
  Serial.print("ppm");
  Serial.print(" ");
  Serial.print("\n");
  Serial.print("Smoke:");
  Serial.print(MQ2Procent2(MQ2Rs(MQ2)/Ro,Smoke));
  Serial.print("ppm");
  Serial.print(" ");
  Serial.print("\n");
}
```

Figur A.4

I loopen ligger även en if-sats som säger till dioden och buzzern vad den ska göra utifrån vilket värde gassensorn upptäcker. Sista delen är funktionerna där beräkning av kalibrering för gassensorn och beräkning av ppm sker.

```
if(MQ2Procent2(MQ2Rs(MQ2)/Ro,Alkohol)>500 or MQ2Procent2(MQ2Rs(MQ2)/Ro,Smoke)>500){
  digitalWrite(redLampa, HIGH);
  digitalWrite(gullLampa, LOW);
  tone(buzzer, 1000, 200);
}
else
{
  digitalWrite(redLampa, LOW);
  digitalWrite(gullLampa, HIGH);
  noTone(buzzer);
}

//-----Funktioner-----//

//Kalibrering av MQ2-sensorn, checkar av så att sensorn är i ren luft.-----//
float MQ2Kalibrering(int port)
{
  int i;
  float val=0;

  for (i=0;i<KalibreringsProver;i++) {
    val += MQ2Resistans(analogRead(port));
    delay(KalibreringsInterval);
  }
  val = val/KalibreringsProver;

  val = val/RorenLuft;

  return val;
}

//Resistans uträkning för MQ2-sensorn-----//
float MQ2Resistans(int port)
{
  return ( ((float)RL*(1023-port)/port));
}
```

Figur A.5

Figur A.6 presenterar slutet av koden för gassensorn där det sista funktionerna presenteras, bland annat uträkning av ppm.

```
//Uträkning av Rs-----  
float MQ2Rs(int port)  
{  
    int i;  
    float rs=0;  
  
    for (i=0;i<ProvInterval;i++) {  
        rs += MQ2Resistans(analogRead(port));  
        delay(AntalProver);  
    }  
  
    rs = rs/ProvInterval;  
  
    return rs;  
}  
  
//Räknar ut PPM av den specifika gasen-----  
int MQ2Procent(float RsDelatRo, float *pcurve)  
{  
    return (pow(10, ((log(RsDelatRo)-pcurve[1])/pcurve[2]) + pcurve[0]));  
}  
  
//Räknar ut ppm av den specifika gasen-----  
int MQ2Procent2(float RsDelatRo, int gas)  
{  
    if ( gas == Alkohol ) {  
        return MQ2Procent(RsDelatRo,AlkoholKurva);  
    }  
    else if ( gas == Smoke ) {  
        return MQ2Procent(RsDelatRo,SmokeKurva);  
    }  
  
    return 0;  
}
```

Figur A.6