HOCHSCHULE SCHMALKALDEN
UNIVERSITY OF APPLIED SCIENCES

Mobile Mustererkennung

Tutorium

Fakultät Informatik

Master

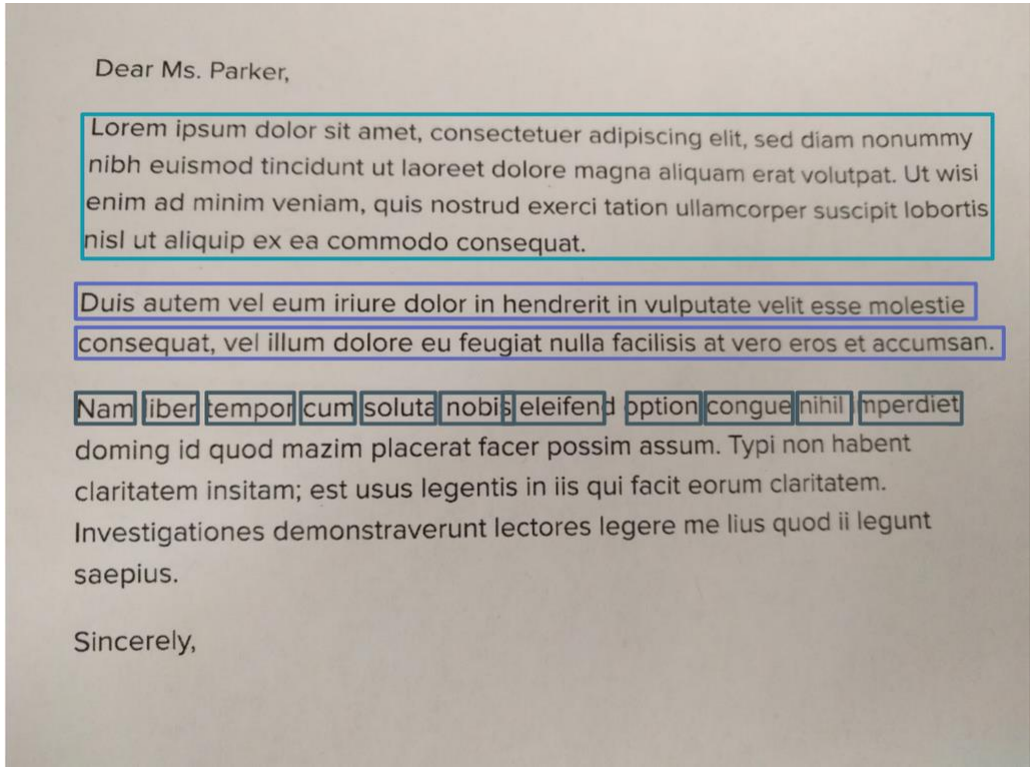
Anwendungsbereiche

Häufigste Anwendung: Erkennung und Analyse von Daten, die über Kamera aufgezeichnet wurden (Bilder, Videos)

- Texterkennung
- Gesichtserkennung
- Landmark - Detektion

Texterkennung

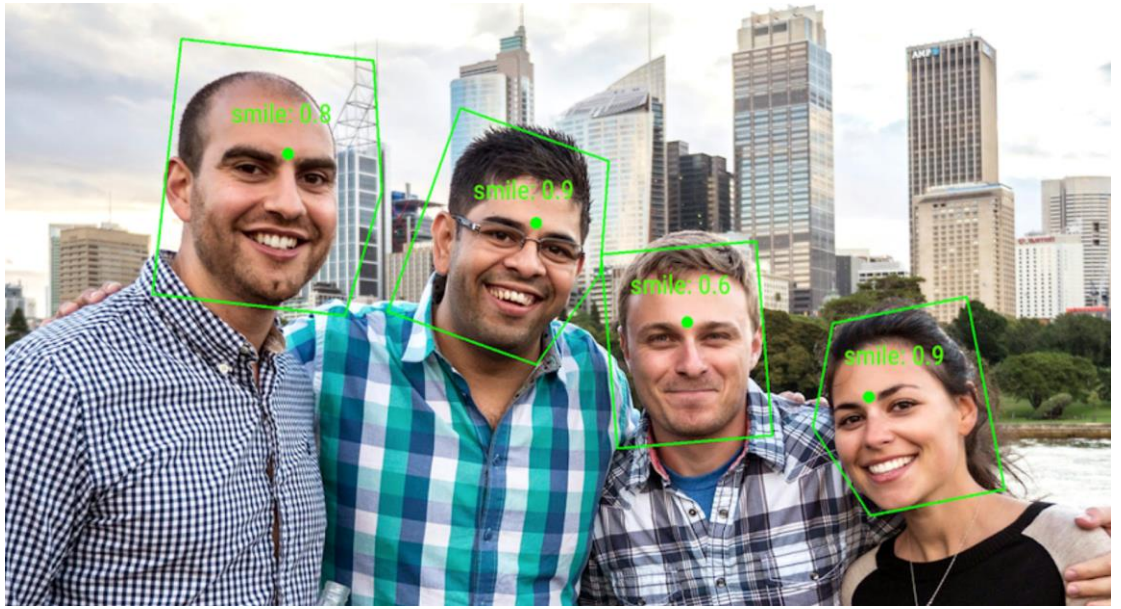
- Verwendung von OCR
- Aufnahme oder Live-Ansicht
- Live Auswertung (Übersetzer) oder weitere Datenverarbeitung zur Kontextgewinnung (Bag of Words)



https://developers.google.com/vision/android/text-overview

Gesichtserkennung

- Aufnahme oder Live-Ansicht
- Gesten Erkennung
  - Mund offen/geschlossen
  - Augen offen/geschlossen
- Erkennung männlich/weiblich



https://developers.google.com/vision/

Google Vision API / Google Firebase

HOCHSCHULE SCHMALKALDEN
UNIVERSITY OF APPLIED SCIENCES

- Vision: API zur Detektion von Text, Gesichtern, Barcodes etc.
  - Nun Bestandteil von Firebase
- Firebase: Bietet Funktionalitäten wie Analyse-Tools und Datenbanken
  - Analytisches Modul: ML Kit

Mobile Vision

Firebase

ML Kit

- Modul für die Integration von Machine Leraning Verfahren durch bereitgestellte Funktionen
- APIs für lokale und Cloudverarbeitung
- Lokal: nicht Netzwerkabhängig
- Netzwerk: höhere Genauigkeit durch Google Cloud Zugriff

App für Texterkennung

- Aufgabe: Schreiben einer Android Applikation, um ein Dokument mit der Kamera aufzunehmen, den Text zu erkennen und ihn in der Applikation ausgeben zu lassen. Der Text soll anschließend durch ein eigens Implementierten Bag of Words Algorithmus laufen, um BoW Merkmale zu gewinnen.
- Verwendung von Google Vision zur Texterkennung
- Schreiben der BoW Klasse

Integration Vision API

- https://codelabs.developers.google.com/codelabs/mobile-vision-ocr/#0

Bag of Words Klasse

```
String str;

List<String> simpleList = new ArrayList<>();
List<String> rawInputList = new ArrayList<>();

List<WordCount> simpleWordCountList = new ArrayList<>();
List<WordCount> nGramWordCountList = new ArrayList<>();

Set<String> stopWordsSet = new HashSet<String>();
```

```
void addToBag(List<String> tmpList) {

    for (int i = 0; i < tmpList.size(); i++) {
        String[] tmp = tmpList.get(i).split( regex: ".");
        rawInputList.addAll(tmpList);
    }
    deleteRawPunctuation();
    Log.d( tag: "BagofWords", msg: "rawInputList: " + String.valueOf(rawInputList));


    List<String> bag = new ArrayList<>();
    for (int i = 0; i < tmpList.size(); i++) {
        String[] tmp = tmpList.get(i).split( regex: " ");
        for (int j = 0; j < tmp.length; j++) {
            bag.add(tmp[j].toLowerCase());
        }
    }

    stopWordsSet.add("a");
    stopWordsSet.add("the");
    stopWordsSet.add("of");
    stopWordsSet.add("is");
    stopWordsSet.add("in");
    stopWordsSet.add("that");
    stopWordsSet.add("as");
    stopWordsSet.add("it");
    stopWordsSet.add("to");

    simpleList.addAll(bag);
    deletePunctuation();
    deleteStopWords();
}
```

HOCHSCHULE SCHMALKALDEN
UNIVERSITY OF APPLIED SCIENCES

```java
private void deleteRawPunctuation() {
    List<String> cleanBag = new ArrayList<>();
    for (int i = 0; i < rawInputList.size(); i++) {
        cleanBag.add(rawInputList.get(i).replace( target: ".",  replacement: "").toLowerCase());
    }

    rawInputList = cleanBag;
}

private void deletePunctuation() {
    List<String> cleanBag = new ArrayList<>();
    for (int i = 0; i < simpleList.size(); i++) {
        cleanBag.add(simpleList.get(i).replace( target: ".",  replacement: ""));
    }
    simpleList = cleanBag;
}
```

```java
private void deleteStopWords() {
    List<String> cleanBag = new ArrayList<>();

    for (int i = 0; i < simpleList.size(); i++) {
        if (!stopWordsSet.contains(simpleList.get(i))) {
            cleanBag.add(simpleList.get(i));
        }
    }
    simpleList = cleanBag;
}

private List<String> deleteStopWordsForNGramModel(String[] stringList) {
    List<String> cleanBag = new ArrayList<>();

    for (int i = 0; i < stringList.length; i++) {
        if (!stopWordsSet.contains(stringList[i])) {
            cleanBag.add(stringList[i]);
        }
    }
    return cleanBag;
}
```

```
void makeSimpleBag() {
    Collections.sort(simpleList);
    if (simpleWordCountList.size() != 0) {
        simpleWordCountList = new ArrayList<>();
    }
    List<String> tmpList = new ArrayList<>(simpleList);
    tmpList.add("");
    int i = 1;
    for (int j = 0; j < tmpList.size() - 1; j++) {

        WordCount wordCount = new WordCount();
        WordCount wordCount1 = new WordCount();

        wordCount.setWord(tmpList.get(j));
        wordCount1.setWord(tmpList.get(j + 1));

        if (wordCount.getWord().equals(wordCount1.getWord())) {
            i++;
            tmpList.remove( i: j + 1);
            j--;
        } else {
            wordCount.setCount(wordCount.getCount() + i);
            i = 1;
            simpleWordCountList.add(wordCount);
        }
    }
    sortByCount(simpleWordCountList);
}
```

```
void makeNGramBag(int N) {
    if (nGramWordCountList.size() != 0) {
        nGramWordCountList = new ArrayList<>();
    }
    List<String> tmpNGramList = new ArrayList<>();
    for (int i = 0; i < rawInputList.size(); i++) {
        String[] tmp = rawInputList.get(i).split( regex: " ");
        List <String> cleanList = deleteStopWordsForNGramModel(tmp);
        for (int j = 0; j < cleanList.size() - N + 1; j++) {
            String tmpNGram = "";
            for (int k = 0; k < N; k++) {
                tmpNGram = tmpNGram + " " + cleanList.get(j + k);
            }
            tmpNGramList.add(tmpNGram);
            Log.d( tag: "BagofWords",  msg: "NgramModel: " + String.valueOf(tmpNGramList));
        }
    }
    Collections.sort(tmpNGramList);
    tmpNGramList.add("");
```

```
    int i = 1;
    for (int j = 0; j < tmpNGramList.size() - 1; j++) {

        WordCount wordCount = new WordCount();
        WordCount wordCount1 = new WordCount();

        wordCount.setWord(tmpNGramList.get(j));
        wordCount1.setWord(tmpNGramList.get(j + 1));

        if (wordCount.getWord().equals(wordCount1.getWord())) {
            i++;
            tmpNGramList.remove( i: j + 1);
            j--;
        } else {
            wordCount.setCount(wordCount.getCount() + i);
            i = 1;
            nGramWordCountList.add(wordCount);
        }
    }
    sortByCount(nGramWordCountList);
}
```

```java
private void sortByCount(List<WordCount> tmpList) {
    for (int i = 0; i < tmpList.size() - 1; i++) {
        if (tmpList.get(i).getCount() > tmpList.get(i + 1).getCount()) {
            WordCount tmp = tmpList.get(i);
            tmpList.set(i, tmpList.get(i + 1));
            tmpList.set(i + 1, tmp);
            i = 0;
        }
    }
}

void clearBag() {
    simpleList = new ArrayList<>();
    simpleWordCountList = new ArrayList<>();

    rawInputList = new ArrayList<>();
}

public List<String> getSimpleList() { return simpleList; }
```

```java
public String toString() {
    if (simpleWordCountList != null) {
        String wordCountListString = "";
        for (int i = 0; i < simpleWordCountList.size(); i++) {
            wordCountListString = wordCountListString + simpleWordCountList.get(simpleWordCountList.size() - i - 1).toString() + "\n";
        }
        return wordCountListString;
    } else {
        return "No simpleWordCountList created yet.";
    }

}

public String toString(int n) {
    if (simpleWordCountList != null) {
        String wordCountListString = "";
        for (int i = 0; i < n; i++) {
            wordCountListString = wordCountListString + simpleWordCountList.get(simpleWordCountList.size() - i - 1).toString() + "\n";
        }
        return wordCountListString;
    } else {
        return "No simpleWordCountList created yet.";
    }

}
```

```java
public String nGramtoString() {
    if (nGramWordCountList != null) {
        String wordCountListString = "";
        for (int i = 0; i < nGramWordCountList.size(); i++) {
            wordCountListString = wordCountListString + nGramWordCountList.get(nGramWordCountList.size() - i - 1).toString() + "\n";
        }
        return wordCountListString;
    } else {
        return "No nGramWordCountList created yet.";
    }

}

public String nGramtoString(int n) {
    if (nGramWordCountList != null) {
        String wordCountListString = "";
        for (int i = 0; i < n; i++) {
            wordCountListString = wordCountListString + nGramWordCountList.get(nGramWordCountList.size() - i - 1).toString() + "\n";
        }
        return wordCountListString;
    } else {
        return "No nGramWordCountList created yet.";
    }

}
```

Verbinden der Komponenten

HOCHSCHULE SCHMALKALDEN
UNIVERSITY OF APPLIED SCIENCES

```
//tts.speak(text.getValue(), TextToSpeech.QUEUE_ADD, null, "DEFAULT");
String string = text.getValue();

Intent myIntent = new Intent( packageContext: this ,TextPreviewActivity.class);
myIntent.putExtra( name: "previewValue",string);
startActivity(myIntent);
```

```java
public class TextPreviewActivity extends AppCompatActivity {

    TextView textView;
    Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_text_preview);

        textView = (TextView)findViewById(R.id.textPreview);
        textView.setText(getIntent().getStringExtra( name: "previewValue"));
        final String string = getIntent().getStringExtra( name: "previewValue");

        button = (Button) findViewById(R.id.button0);
        button.setOnClickListener(new View.OnClickListener() {

            public void onClick(View view) {

                Intent myIntent = new Intent(view.getContext(),BagOfWordsActivity.class);
                myIntent.putExtra( name: "BagOfWords",string);
                startActivity(myIntent);

            }
        });
    }
}
```

```java
public class BagOfWordsActivity extends AppCompatActivity {

    List<String> value = new ArrayList<>();
    BagOfWords bag = null;
    TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bag_of_words4);

        textView = (TextView)findViewById(R.id.bagOfWordsText);

        value.add(getIntent().getStringExtra( name: "BagOfWords"));
        bag = new BagOfWords();
        bag.addToBag(value);
        bag.makeSimpleBag();
        Log.d( tag: "BagOfWordsActivity",  msg: "Bag: " + String.valueOf(bag.toString()));

        bag.makeNGramBag( N: 2);

        textView.append(bag.toString( n: 10));
        textView.append("---------------------- \n");
        textView.append(bag.nGramtoString( n: 10));
    }
}
```