

Künstliche Intelligenz, Machine and Deep Learning

Lab-1.2 Microsoft Machine Learning
Azure Cognitive Services



Mykola
Dobrochynskyy
ceo@soft-fact.de

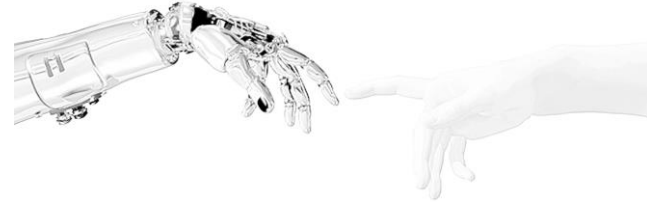


Agenda



Agenda

1. Künstliche Intelligenz - Einführung
2. Machine Learning - Grundlagen
3. Machine Learning - klassische Methoden
4. Künstliche Neuronale Netze
5. Deep Learning für Computer Vision
6. Deep Learning für Texte und Sequenzen
7. Generatives Deep Learning, Reinforcement Learning, Adversarial und Quantum Machine Learning und andere Forschung- und Zukunftsthemen
8. KI Infrastruktur - Ressourcen und Dokumentation, Hardware, Bibliotheken, KI Cloud-Dienste
9. Demo Lab-1 - 1.1 - Azure Machine Learning Studio und **1.2 – Azure Cognitive Services**
10. Demo Lab-2 - Machine Learning Services von Amazon AWS
11. Demo Lab-3 - Machine Learning Services von Google Cloud
12. Demo Lab-4 – Bild-Klassifizierung mit CNN in Python



9. Demo Lab 1.2

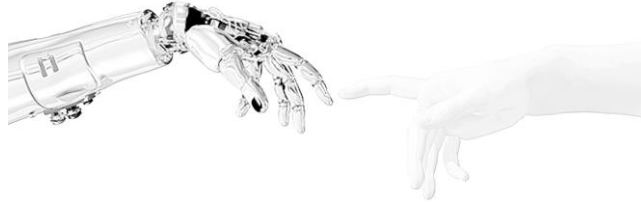
Microsoft Machine Learning

Inhalt



Inhalt

1. **Microsoft Machine Learning – Übersicht** –
Folien 4. – 6.
2. **Azure Cognitive Services – Übersicht** –
Folien 7. – 13.
3. **Azure Cognitive Services. Maschinelles Sehen –**
exemplarisches Erkunden und Testen via Web-Portal –
Folien 14. – 27.
4. **Beispiel-Projekte: Bildanalyse & OCR** –
Folien 28. – 33.
5. **Beispiel-Projekte: Textanalyse** – Folien 34. – 35.
6. **Abschluss** (weitere Schritte, Ressourcen & Links,
Kontakt-Info) – Folien 36. – 38.
7. **API-Keys von Cognitive Services in Visual Studio**
– Folie N° 39



9. Demo Lab 1.2

Microsoft Machine Learning Übersicht

Microsoft Machine Learning. Cloud-basierte Optionen

Cloudoptionen	Funktionsbeschreibung	Gebotene Möglichkeiten
Azure Machine Learning Service	Verwalteter Clouddienst für ML	Trainieren, Bereitstellen und Verwalten von Modellen in Azure mithilfe von Python und CLI
Azure Machine Learning Studio (im Lab 1.1 vorgestellt)	Visuelle Drag-&-Drop-Oberfläche für ML	Erstellen von, Experimentieren mit und Bereitstellen von Modellen mithilfe von vorkonfigurierten Algorithmen (Python und R)
Azure Databricks	Spark-basierte Analyseplattform	Erstellen und Bereitstellen von Modellen und Datenworkflows
Azure Cognitive Services (hier im Lab 1.2 vorgestellt)	Azure-Dienste mit vorkonfigurierten KI- und ML-Modellen	Fügen Sie Ihren Apps auf einfache Weise intelligente Features hinzu
Azure Data Science Virtual Machine	Virtueller Computer mit vorinstallierten Data Science-Tools	Entwickeln von ML-Lösungen in einer vorkonfigurierten Umgebung



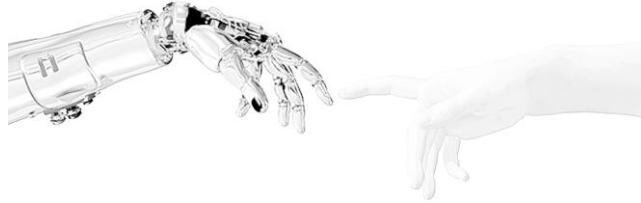
9. Demo Lab 1.2

Microsoft Machine Learning

Übersicht

Microsoft Machine Learning. Lokale Optionen

Lokale Optionen	Funktionsbeschreibung	Gebotene Möglichkeiten
<u>SQL Server Machine Learning Services</u>	In SQL eingebettete Analyse-Engine	Erstellen und Bereitstellen von Modellen innerhalb von SQL Server
<u>Microsoft Machine Learning Server</u>	Eigenständiger Enterprise-Server für prädiktive Analyse	Erstellen und Bereitstellen von Modellen mit R und Python



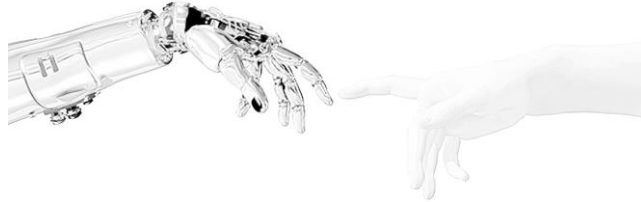
9. Demo Lab 1.2

Microsoft Machine Learning

Übersicht

Microsoft Machine Learning. Entwicklungstools

Entwicklungstools	Funktionsbeschreibung	Gebotene Möglichkeiten
<u>ML.NET</u>	Plattform-übergreifendes Open-Source Machine Learning SDK	Entwickeln von ML-Lösungen für .NET-Anwendungen
<u>Infer.NET</u>	Plattformübergreifendes Open-Source Machine Learning Framework (wird in ML.NET integriert)	Bayessche Inferenz in graphischen Modellen und probabilistische Programmierung mit C# auf .NET und .NET Core Plattformen
<u>Windows ML</u>	Windows 10 Machine Learning Plattform	Auswerten von trainierten Modellen auf einem Windows 10-Gerät



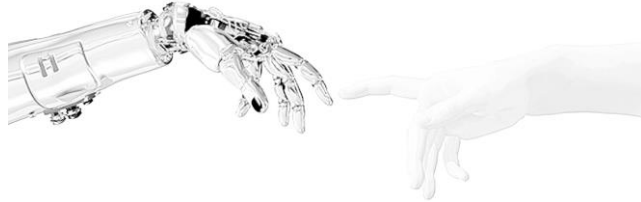
9. Demo Lab 1.2

Azure Cognitive Services

Übersicht

Azure Cognitive Services





9. Demo Lab 1.2

Azure Cognitive Services

Bildanalyse



Azure Cognitive Services. Bildanalyse



Maschinelles Sehen

- Bildklassifizierung,
- [Szenen- und Aktivitäts-erkennung](#),
- [Erkennung von Berühmtheiten und Wahrzeichen](#),
- [Optische Zeichen-erkennung \(OCR\)*](#),
- [Handschrift-erkennung](#)



Video- Indizierung

- Gesichtserkennung in Videos
- Objekt-, Szenen- und Aktivitäts-erkennung in Videos
- Metadaten-, Audio- und Schlüssel-Extraktion und Analyse



Gesichts- erkennung

- [Gesichts-erkennung*](#)
 - Personen-identifizierung
 - [Emotions-erkennung*](#)
 - [Erkennung und Gruppierung ähnlicher Gesichter](#)
-
- [siehe Artikel und Demos](#)



Content Moderator

- Moderation von anstößigen Inhalten in Bildern / Videos,
- Benutzerdefinierte Abbildung- und Textlisten zum Blockieren oder Zulassen,
- Einbinden von menschlichen Moderatoren.

[Siehe Artikel und Demos](#)



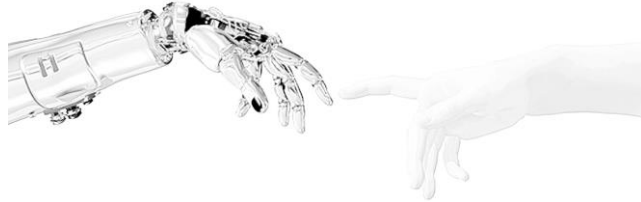
Custom Vision

[Benutzerdefinierte und anpassbare Bild-erkennung](#)

Tipp: Erstellen Sie Ihr Bild-Klassifizierungs-Projekt mit Ihren eigenen Bildern auf customvision.ai

[Anleitung siehe hier.](#)

* - Unterstützt Docker-Container



9. Demo Lab 1.2

Azure Cognitive Services

Spracheingabe

Azure Cognitive Services. Spracheingabe



Sprach- erkennung

- Automatische und anpassbare Spracherkennung und Transkription (Sprache-in-Text)
- Anpassbare Sprachmodelle für eine einzigartige Sprachvielfalt oder Akzente



Text in Sprache umwandeln

- Automatische Sprachsynthese
- Anpassbare Voicefonts für die Sprachsynthese



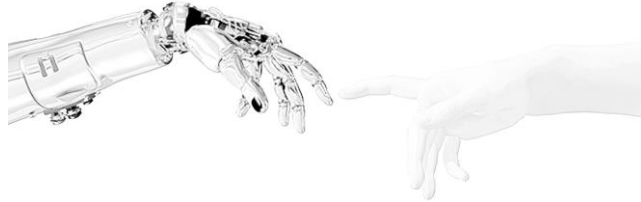
Sprecher- erkennung

- Sprecher-Identifizierung
- Sprecher-Überprüfung



Sprach- übersetzung

- Echtzeit-übersetzung
- Automatisierte Sprach-übersetzung
- Anpassbare Benutzer-definierte Sprach-übersetzungssysteme erstellen



9. Demo Lab 1.2

Azure Cognitive Services

Sprache

Azure Cognitive Services. Sprache



Text- Analysen

- Eigennamen-
erkennung

- Stichwörter
extrahieren*

- Standpunkt-
analyse von
Texten*



Translator Text

- Automatische
Sprachenerkenn-
ung*

- Automatisierte
Textübersetzung

- Anpassbare
Übersetzung



Language Understanding (LUIS)

- Kontext-
bezogene
Sprachanalyse



Content Moderator

- Moderation
von anstößigen
Inhalten in
Bildern / Videos,
- Benutzer-
definierte
Abbildung- und
Textlisten zum
Blockieren oder
Zulassen,
- Einbinden von
menschlichen
Moderatoren.



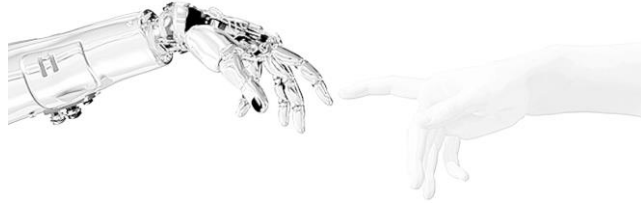
Bing- Rechtschreib- prüfung

- Webbasierte
mehrsprachige
Rechtschreib-
prüfung

Kontext-
bezogene
Rechtschreib-
prüfung

* - Unterstützt Docker-Container

[Siehe Artikel und
Demos](#)



9. Demo Lab 1.2

Azure Cognitive Services

Einblicke und Wissen

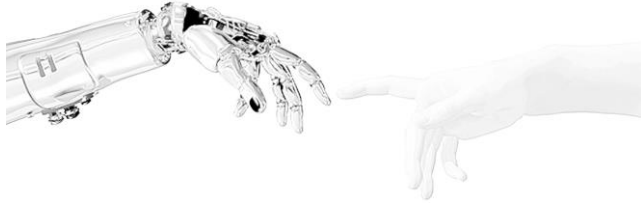
Azure Cognitive Services. Einblicke und Wissen



QnA Maker

[Kostenlos testen](#) | [Weitere Informationen](#)

- [Extraktion von Fragen und Antworten aus unstrukturiertem Text](#)
- [Erstellung von Wissensdatenbanken aus Q&A-Sammlungen](#)
- Semantische Zuordnung für Wissensdatenbanken



9. Demo Lab 1.2

Azure Cognitive Services

Suche (Teil 1.)

Azure Cognitive Services. Suche



- **Bing-Websuche**

[Kostenlos testen](#) | [Weitere Informationen](#)

Werbefreie Websuche | [Demo](#)

Sichere Websuche

Standortbezogene Websuche



- **Visuelle Bing-Suche**

[Kostenlos testen](#) | [Weitere Informationen](#)

Identifizierung ähnlicher Bilder und Produkte | [Demo](#)

Wissensbeschaffung aus Bildern

Identifizierung von Webquellen



- **Benutzerdefinierte Bing-Suche**

[Weitere Informationen](#)

Erstellung von benutzerdefinierten Suchmaschinen

Werbefreie benutzerdefinierte Suchergebnisse | [Demo](#)



- **Bing-Entitätssuche**

[Kostenlos testen](#) | [Weitere Informationen](#)

Erkennung und Klassifizierung von Eigennamen | [Demo](#)

Wissensbeschaffung für Eigennamen



- **Bing-Videosuche**

[Kostenlos testen](#) | [Weitere Informationen](#)

Werbefreie Videosuche | [Demo](#)

Identifizierung von Videothemen und Trends



9. Demo Lab 1.2

Azure Cognitive Services

Suche (Teil 2.)

Azure Cognitive Services. Suche



- **Bing-News-Suche**

[Kostenlos testen](#) | [Weitere Informationen](#)
Werbefreie News-Suchergebnisse | [Demo](#)
Identifizierung von Trendthemen



- **Bing-Bildersuche**

[Kostenlos testen](#) | [Weitere Informationen](#)
Werbefreie Bildersuche | [Demo](#)



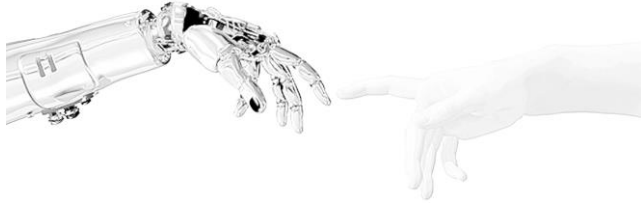
- **Bing-Vorschlagssuche**

[Kostenlos testen](#) | [Weitere Informationen](#)
Automatische Vervollständigung von Suchabfragen | [Demo](#)



- **Bing-Suche für ortsansässige Unternehmen** ^{VORSCHAU}

[Kostenlos testen](#) | [Weitere Informationen](#)
Finden lokaler Unternehmen im Interessenbereich | [Demo](#)



9. Demo Lab 1.2

Azure Cognitive Services



Maschinelles Sehen

Bildanalyse

Testen im Web

Software Factories
makes software that works



Dieses Feature gibt Informationen zu visuellen Inhalten in einem Bild zurück. Man kann Tagging, domänenspezifische Modelle und Beschreibungen nutzen, um Inhalte zu identifizieren. Mann kann auch potenziell nicht jugendfreie Inhalte damit erkennen sowie die Bildtypen und Farbschemas in Bildern herausfinden.

Test-Anweisungen:

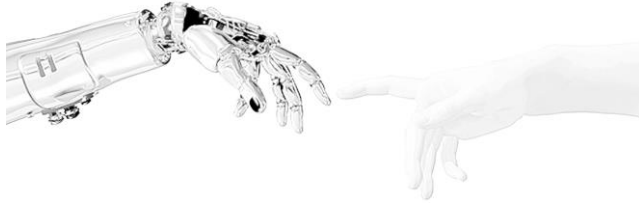
1. [„Maschinelles Sehen“-Link](#) im Browser öffnen.
2. Zum „Bilder analysieren“-Abschnitt gehen (s. Bild unten).
3. Verschiedene fertige Bilder anklicken und abgefragte Bild-Informationen analysieren (graues Kästchen oben rechts).
4. Alternativ ein Bild zur Analyse vom lokalen Rechner hochladen (Button „Durchsuchen“ klicken und Datei auswählen) oder aus dem Internet (Bild-Adresse im „Bild-URL“-Textbox angeben und auf „Senden“ klicken).

NAME DES MERKMALS:	WERT
Beschreibung	{ "tags": ["train", "platform", "station", "building", "indoor", "subway", "track", "walking", "waiting", "pulling", "board", "people", "man", "luggage", "standing", "holding", "large", "woman", "yellow", "suitcase"], "captions": [{ "text": "people waiting at a train station", "confidence": 0.8330993 }] }
Tags	[{ "name": "train", "confidence": 0.9975446 }, { "name": "platform", "confidence": 0.995543063 }, { "name": "station", "confidence": 0.9798007 }, { "name": "indoor", "confidence": 0.9277198 }, { "name": "subway", "confidence": 0.838939548 }, { "name": "pulling", "confidence": 0.4317156 }]
Bildformat	"Jpeg"
Bildmaße	162 x 600

Möchten Sie dies erstellen?

Weitere Informationen >

Hier klicken, um Schnellstart-Doku zu öffnen



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

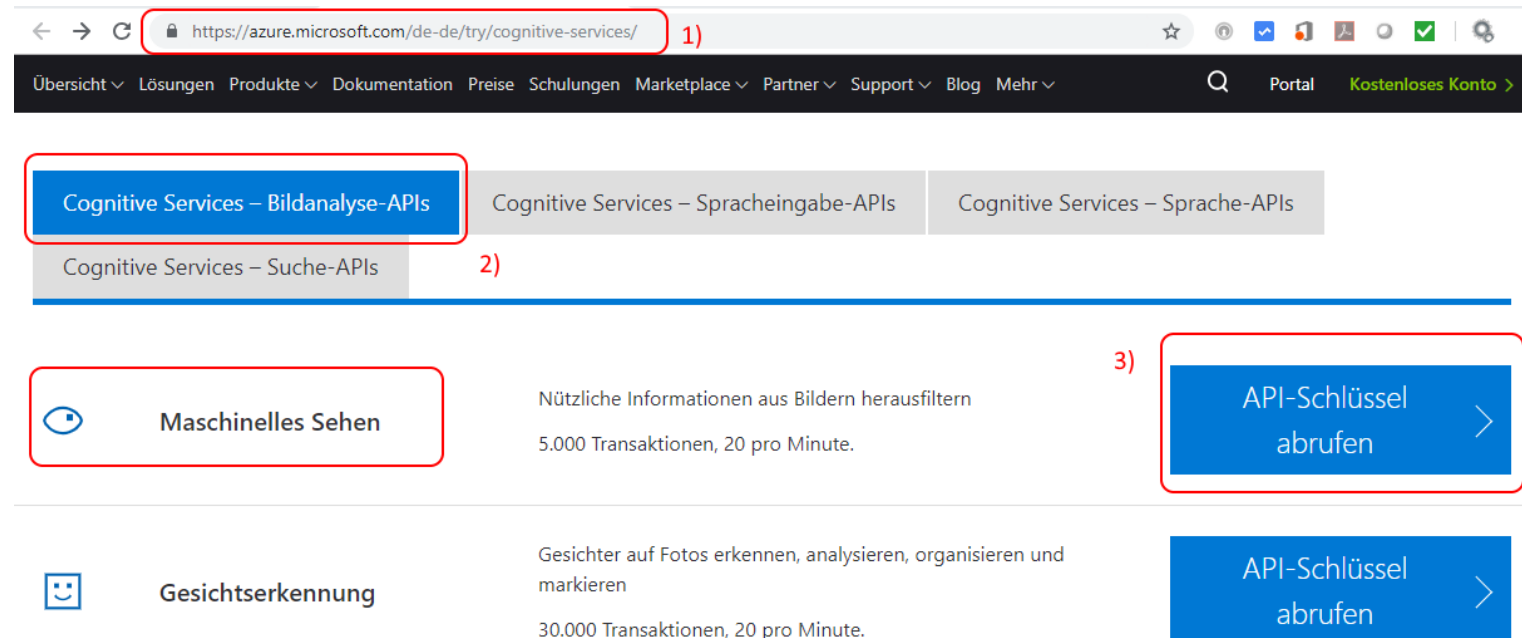
REST-API Keys abfragen



Um ein Azure Cognitive Dienst nutzen zu können, muss man zuerst ein Dienst erstellen (als 7-tätige Gast Test-Version oder mit einem neuen kostenlosen oder vorhandenen Azure-Konto) und können die API-Schlüssel abgefragt werden, die für den Zugriff aus einem Client-Programm benötigt sein werden. In den Folien „REST-API Keys abfragen“ wird zwar der Fall mit „Maschinelles Dienst“ beschrieben, die gleiche Vorgehensweise kann aber analogisch auch für andere Azure Cognitive Dienste verwendet werden.

Anweisungen:

1. „[Cognitive Services ausprobieren](https://azure.microsoft.com/de-de/try/cognitive-services/)“-Link im Browser öffnen.
2. „Cognitive Services – Bildanalyse-APIs“-Reiter auswählen
3. Auf den Button „API-Schlüssel abrufen“ drücken (s. Bild unten)
4. Weiteren Hinweisen auf der nächsten Folie folgen





Übersicht ▾ Lösungen Produkte ▾ Dokumentation Preise Schulungen Marketplace ▾ Partner ▾ Support ▾ Blog Mehr ▾

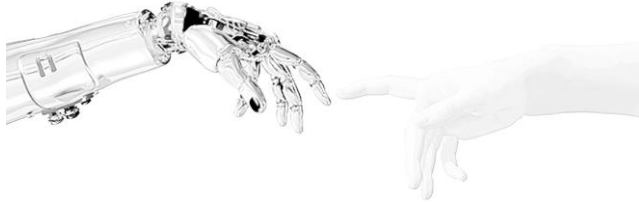
Portal **Kostenloses Konto >**

Cognitive Services – Bildanalyse-APIs Cognitive Services – Spracheingabe-APIs Cognitive Services – Sprache-APIs

Cognitive Services – Suche-APIs 2)

 **Maschinelles Sehen** Nützliche Informationen aus Bildern herausfiltern
5.000 Transaktionen, 20 pro Minute. 3) **API-Schlüssel abrufen >**

 **Gesichtserkennung** Gesichter auf Fotos erkennen, analysieren, organisieren und markieren
30.000 Transaktionen, 20 pro Minute. **API-Schlüssel abrufen >**



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

REST-API Keys abfragen

Testzugang auswählen



Es gibt drei Test-Optionen (s. Bild unten):

- 7-Tage-Testversion (für einen kurzen Test, keine Kreditkarte erforderlich) – s. Folien N° 17
- Kostenloses Azure-Konto (170€ Gutschrift, gleich kostenlos starten) – s. Folie N° 18 - 20
- Vorhandenes Azure-Konto (wenn man bereits Azur-Konto hat) – s. Folie N° 18 - 20

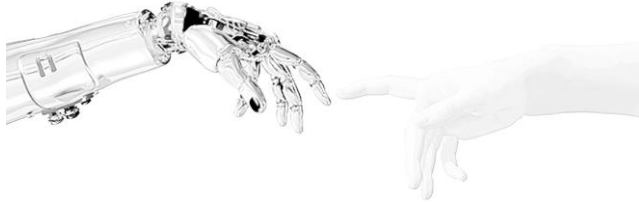
Wählen Sie eine passende Option und drücken Sie auf jeweiligen Button in der Mitte.

×

Cognitive Services kostenlos testen

Gast	Kostenloses Azure-Konto	Vorhandenes Azure-Konto
7-Tage-Testversion Cognitive Services kostenlos ausprobieren Erste Schritte	€0/Monat Selbst ausprobieren – mit einem kostenlosen Azure-Konto Registrieren	Haben Sie bereits ein Azure-Konto? Anmelden
<ul style="list-style-type: none">• Keine Kreditkarte erforderlich• Nach der Testversion werden keine Daten gespeichert	<ul style="list-style-type: none">• Kostenlose Pläne, die nie ablaufen• Gutschrift über €170 auf Azure erhalten• Gespeicherte Daten und Anpassungen	<ul style="list-style-type: none">• Vollständiger SLA-Bericht• Vollständige Azure-Integration• Nahtlos skalieren

Es wird **empfohlen kostenloses oder vorhandenes Azure-Konto zu nutzen**, da Sie nicht ablaufen und dort mehr Möglichkeiten und Features zum Testen und Evaluieren angeboten werden (s. Folien N° 21 bis 27).



9. Demo Lab 1.2

Azure Cognitive Services

 Maschinelles Sehen

Bildanalyse

REST-API Keys abfragen

7-Tage-Testversion




Anweisungen:

1. Im Dialog „Microsoft Cognitive Services Bedingungen“ stimmen Sie allen Option zu, wählen Sie „Deutschland“ als Land/Region aus und drücken Sie „Nächste“-Button.
2. Aus der nachfolgenden Status-Seite kopieren Sie die Webdienst-Adressen („Endpunkte“ – s. blaues Kästchen im Bild unten) und API-Schlüssel in einen abgesicherten Ort für die Weiterverwendung in Ihrem Programm.
3. Anweisungen ab der Folie N° 28 folgen (Folien N° 18 bis 27 kann man überspringen, wenn man nur 7-Tage-Testversion ausprobieren möchte).

Achtung! Bewahren Sie die API-Schlüssel in einem sicheren Ort und schließen Sie den Zugang der Dritten zu Ihren API-Schlüssel aus. Da sonst den Missbrauch von Dritten die unnötiges Kosten auf Ihr Azure-Konto gebucht werden könnten.

Status Hallo! (Abmelden)



Maschinelles Sehen

Nützliche Informationen aus Bildern herausfiltern

5.000 Transaktionen, 20 pro Minute.

Dieser API-Schlüssel ist zurzeit aktiv.

6 Tage verbleibend

Endpunkte

<https://westcentralus.api.cognitive.microsoft.com/vision/v1.0>

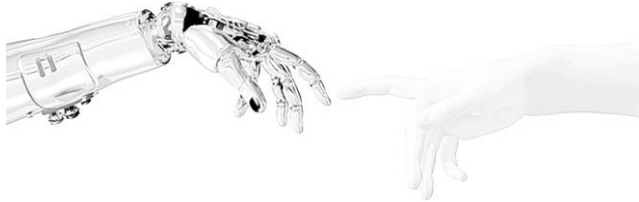
<https://westcentralus.api.cognitive.microsoft.com/vision/v2.0>

Schlüssel 1: 57c15101-1111-4403-b011-000000000000

Schlüssel 2: e7157101-1111-4403-b011-000000000001

API-Schlüssel

[Schnellstarthandbuch >](#)



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

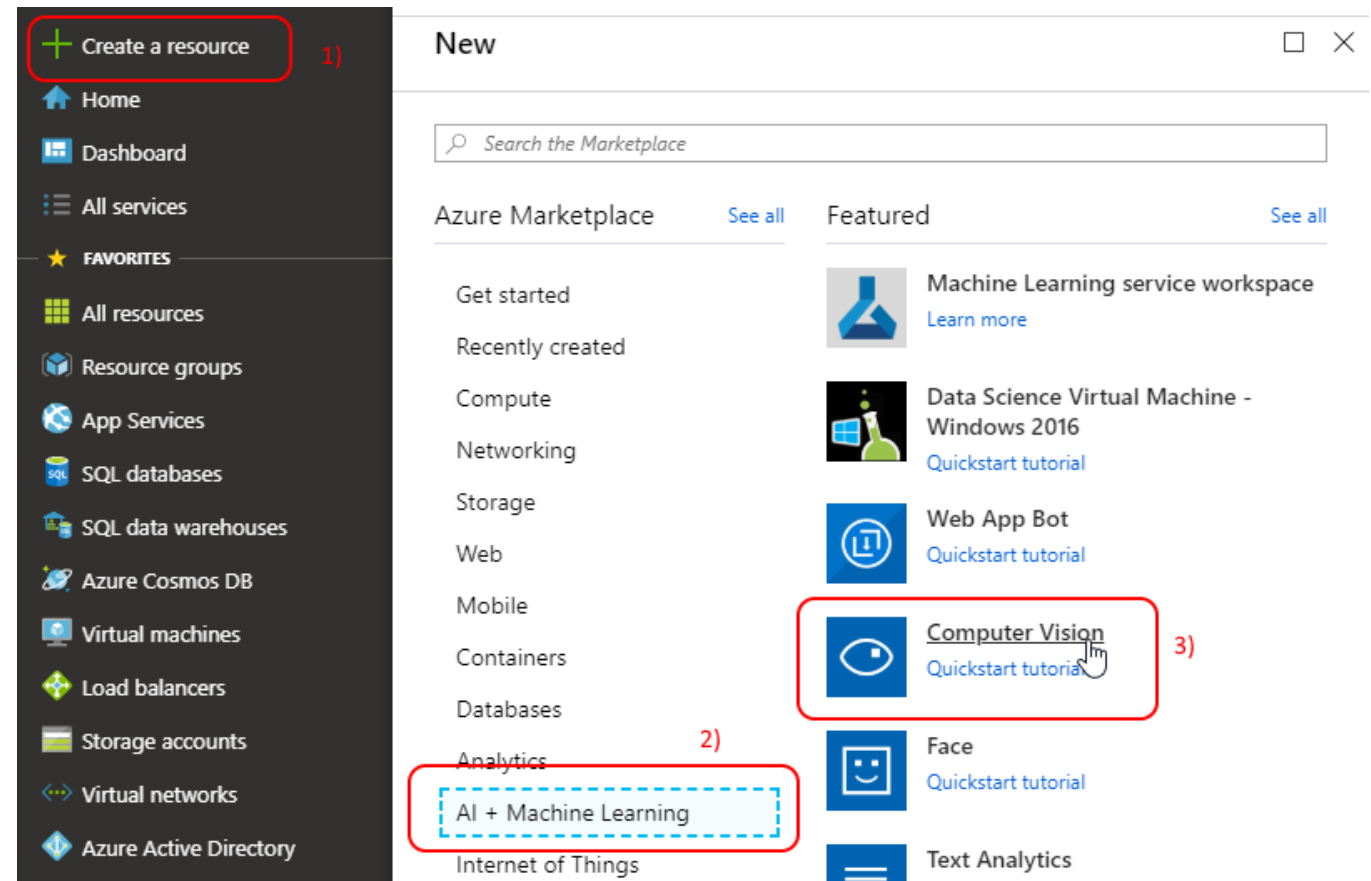
Dienst erstellen

Kostenloses bzw.
vorhandenes Azure-Konto



Anweisungen:

1. Falls Sie für ein [kostenloses Azure-Konto](#) gewählt wurde (s. Button „Registrieren“ auf Folie N° 16) die Registrierungsdaten ausfüllen und weiteren Anweisungen auf dem Bildschirm folgen.
2. Zu <https://portal.azure.com> gehen
3. Auf „Create a resource“ klicken dann auf „AI + Machine Learning“ und auf „Computer Vision“ klicken (s. Bild)





9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

Dienst erstellen

Kostenloses bzw.
vorhandenes Azure-Konto



Home > New > Create

Create

Computer Vision

* Name 1)
Computer_Vision_AI_Service ✓

* Subscription 2)
Free Trial ▾

* Location 3)
East US ▾

* Pricing tier (View full pricing details) 4)
F0 (20 Calls per minute, 5K Calls per month) ▾

* Resource group 5)
(New) AI-Services ▾
[Create new](#)

6)

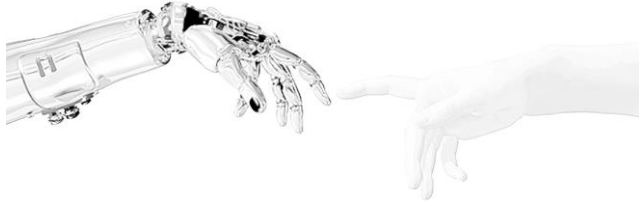
[Create](#) [Automation options](#)

Anweisungen:

1. Dienstname angeben
2. Abo-Typ (Subscription) wählen - z.B. „Free trial“ (kostenlos ausprobieren)
3. Dienst-Ort* (Location) auswählen.
4. Preis-Option** auswählen.
5. Eine vorhandene Ressourcen-Gruppe auswählen oder eine neue erstellen – z.B. AI-Services.
6. Auf „Create“-Button drücken – damit wird der „Computer Vision“-Dienst erstellt.

* - von Dienst-Ort (Location) hängt auch Webdienst-Webadresse (Endpoint-URL) ab

** - bis zu 5000 Aufrufe/Monat mit einer Frequenz bis 20 Aufrufe/Min. sind kostenlos.



9. Demo Lab 1.2

Azure Cognitive Services

 Maschinelles Sehen

Bildanalyse

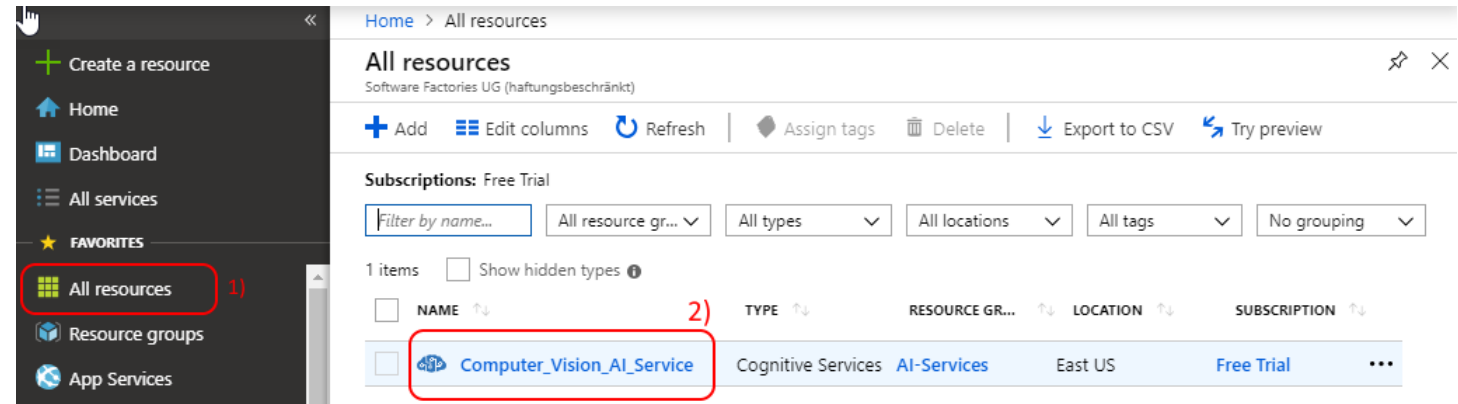
REST-API Keys abfragen

Kostenloses bzw.
vorhandenes Azure-Konto



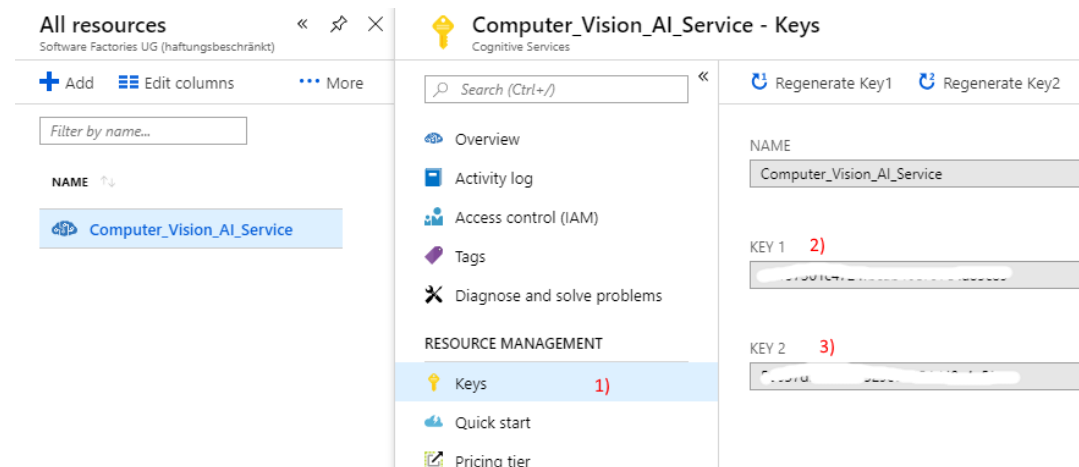
Anweisungen:

1. Zu <https://portal.azure.com> gehen
2. Auf soeben erstellten KI-Dienst klicken (s. Bild unten)
3. Weiteren Anweisungen (s. unten) folgen

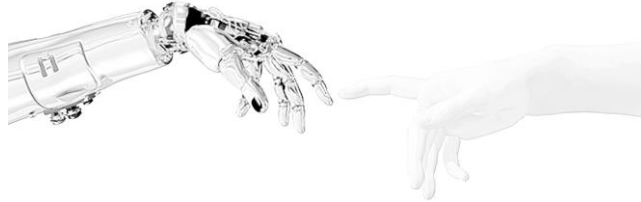


Anweisungen:

1. Auf „Keys“ im Ressource Management klicken (s. Bild).
- 2.-3. „KEY 1“- und „KEY2“-Werte im sicheren Ort speichern.



Achtung! Bewahren Sie die API-Schlüssel in einem sicheren Ort und schließen Sie den Zugang der Dritten zu Ihren API-Schlüssel aus. Da sonst den Missbrauch von Dritten die unnötiges Kosten auf Ihr Azure-Konto gebucht werden könnten.



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

Schnellstart

Kostenloses bzw.
vorhandenes Azure-Konto

Software Factories
makes software that works 

Anweisungen:

1. Zu <https://portal.azure.com> gehen
2. Auf soeben erstellten KI-Dienst und auf „Quick start“ klicken (s. Bild unten)
3. Dienstadresse merken/speichern (s. rot markiert auf dem Bild unten)
4. Schnellstart-Hinweisen in 1 bis 3 (s. Bild) und/oder Anweisungen ab der nachfolgenden Folie N° 22 folgen.

Computer_Vision_AI_Service - Quick start

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

RESOURCE MANAGEMENT


- Keys
- Quick start**
- Pricing tier
- Billing By Subscription
- Properties
- Locks
- Automation script

Monitoring

- Alerts
- Metrics
- Diagnostic settings
- Logs

Support + troubleshooting

- Resource health
- New support request

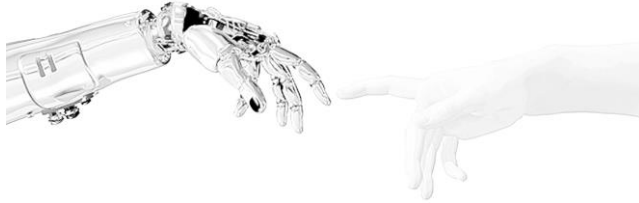
 Congratulations! Now explore the Quickstart guidance to get up and running with Computer Vision.
[Checkout new support for Docker containers in Azure Cognitive Services \(PREVIEW\)](#)

1 Grab your keys
Every web API call of Computer Vision and every Docker container activation of Recognize Text requires a subscription key. For the web API this key needs to be either passed through a query string parameter or specified in the request header. For the Docker container the key needs to be passed through the Docker command.
[Keys](#)

2a Run the Docker container (PREVIEW)
The Recognize Text portion of the Computer Vision Cognitive Service is also available as a Docker container that you can pull and use directly as part of your application. It can detect text in an image using optical character recognition (OCR) and extract the recognized words into a machine-readable character stream. Analyze images to detect embedded text, generate character streams, and enable searching.
[Container Support in Azure Cognitive Services](#)
[Recognize Text container](#)

2b Or make a API call to this endpoint: <https://eastus.api.cognitive.microsoft.com/>
Get in-depth information about each properties and methods of the API. Test your keys with the built-in testing console without writing a single line of code. Once you have the API running, you can check your consumption and the API health on Azure portal in your API 'Overview'.
[Computer Vision API reference](#)
[Realtime API usage](#)
[API metrics alert](#)
[Diagnostics settings](#)
[Logs](#)
[Billing by subscription](#)
[Resource health status](#)

3 Enjoy coding
Learn more about the features, tutorials, developer tools, examples and how-to guidance to speed up.
[Documentation](#)
[SDK](#)
[Samples for Recognize Text container](#)



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

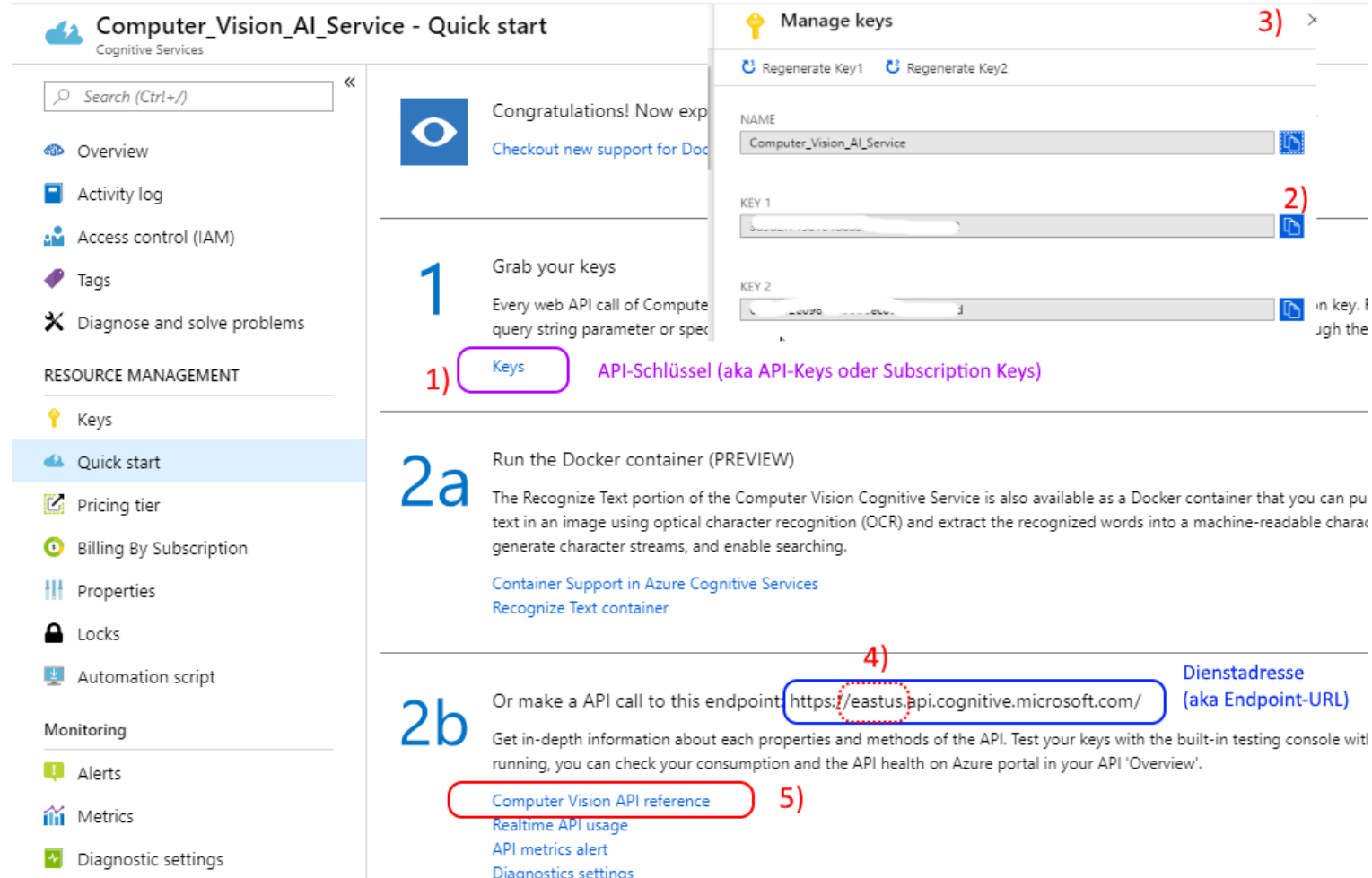
API direkt testen (1)

auf Azure-Portal



Anweisungen (s. Bild):

1. Auf „Quick start“-Seite auf „Keys“-Link drücken (oder auf Keys-Icon links).
2. KEY1 (oder KEY2) in Zwischenablage speichern.
3. „Manage keys“ Fenster mit „X“-Icon schließen.
4. Dienst-Ort merken (auf dem Beispiel-Bild es ist „eastus“).
5. Auf Link „Computer Vision API reference“ drücken.



The screenshot displays the Azure Portal interface for the Computer Vision AI Service. On the left, the navigation pane shows the 'Quick start' section selected. The main content area displays the 'Quick start' page, which includes a 'Grab your keys' section (1) and a 'Run the Docker container (PREVIEW)' section (2a). The 'Grab your keys' section shows a 'Keys' link (1) and a 'Manage keys' window (3) where the 'KEY 1' (2) is displayed. The 'Run the Docker container (PREVIEW)' section shows a 'Container Support in Azure Cognitive Services' link and a 'Recognize Text container' link. The '2b' section shows an API endpoint (4) and a 'Computer Vision API reference' link (5).

Computer_Vision_AI_Service - Quick start

Search (Ctrl+/)

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

RESOURCE MANAGEMENT

Keys
Quick start
Pricing tier
Billing By Subscription
Properties
Locks
Automation script

Monitoring

Alerts
Metrics
Diagnostic settings

1 Grab your keys

Every web API call of Compute query string parameter or spec

1) **Keys** API-Schlüssel (aka API-Keys oder Subscription Keys)

2a Run the Docker container (PREVIEW)

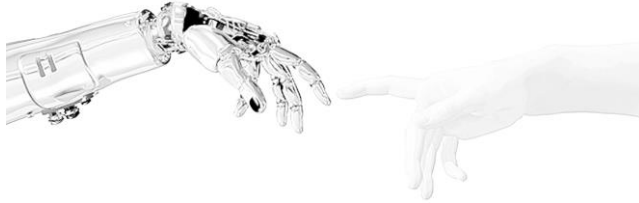
The Recognize Text portion of the Computer Vision Cognitive Service is also available as a Docker container that you can put text in an image using optical character recognition (OCR) and extract the recognized words into a machine-readable character stream, and enable searching.

Container Support in Azure Cognitive Services
Recognize Text container

2b Or make a API call to this endpoint **4)** <https://eastus.api.cognitive.microsoft.com/> **Dienstadresse (aka Endpoint-URL)**

Get in-depth information about each properties and methods of the API. Test your keys with the built-in testing console with running, you can check your consumption and the API health on Azure portal in your API 'Overview'.

5) **Computer Vision API reference**
Realtime API usage
API metrics alert
Diagnostics settings



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

API direkt testen (2)

API-Referenz - Geo-Zonen



Anweisungen (s. Bild):

1. Alle API-Referenzen durchklicken und Dokumentation durchlesen (s. 1a).bis 1j. auf dem Bild). Da sind Request-URLs für APIs in verschiedenen Geo-Zonen, Request-Parameter/-Headers/-Body sowie kurze Code-Beispiele für REST-Abfragen in C#, Java, JavaScript, ObjC, PHP, Python, Ruby und Curl dokumentiert.
2. Bei „Open API testing console“ auf zur Geo-Zone passenden Button drücken – zum Beispiel für US-Ostküste auf „East US“ (s. Bild)

Microsoft
Cognitive Services

APIs Documentation > API Reference

- POST Analyze Image 1a)
- POST Describe Image 1b)
- GET Get Handwritten Text Operation Result 1c)
- POST Get Thumbnail 1d)
- GET List Domain Specific Models 1e)
- POST OCR 1f)
- POST Recognize Domain Specific Content 1g)
- POST Recognize Handwritten Text 1h)
- POST Tag Image 1j)

Computer Vision API - v1.0

The Computer Vision API provides state-of-the-art algorithms to process images and return information. For example, it can be used to determine if an image contains mature content, or it can be used to find all the faces in an image. It also has other features like estimating dominant and accent colors, categorizing the content of images, and describing an image with complete English sentences. Additionally, it can also intelligently generate images thumbnails for displaying large images effectively.

This API is currently available in:

- West US - westus.api.cognitive.microsoft.com
- West US 2 - westus2.api.cognitive.microsoft.com
- East US - eastus.api.cognitive.microsoft.com
- East US 2 - eastus2.api.cognitive.microsoft.com
- West Central US - westcentralus.api.cognitive.microsoft.com
- South Central US - southcentralus.api.cognitive.microsoft.com
- West Europe - westeurope.api.cognitive.microsoft.com
- North Europe - northeurope.api.cognitive.microsoft.com
- Southeast Asia - southeastasia.api.cognitive.microsoft.com
- East Asia - eastasia.api.cognitive.microsoft.com
- Australia East - australiaeast.api.cognitive.microsoft.com
- Brazil South - brazilsouth.api.cognitive.microsoft.com
- Canada Central - canadacentral.api.cognitive.microsoft.com
- Central India - centralindia.api.cognitive.microsoft.com
- UK South - uksouth.api.cognitive.microsoft.com
- Japan East - japaneast.api.cognitive.microsoft.com

Describe Image

This operation generates a description of an image in human readable language with complete sentences. The description is based on a collection of content tags, which are also returned by the operation. More than one description can be generated for each image. Descriptions are ordered by their confidence score. All descriptions are in English.

Two input methods are supported -- (1) Uploading an image or (2) specifying an image URL.

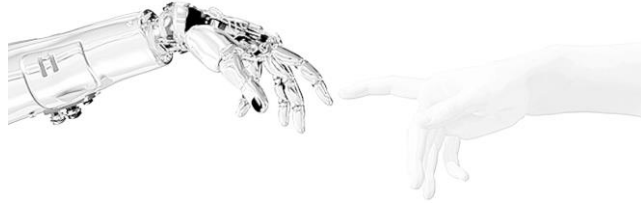
A successful response will be returned in JSON. If the request failed, the response will contain an error code and a message to help understand what went wrong.

Http Method
POST

Open API testing console

2)

West US West US 2 East US East US 2 West Central US South Central US West Europe North Europe Southeast Asia East Asia Australia East Brazil South Canada Central Central India UK South



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

API direkt testen (3)

API-Referenz



Abfrage-Parameter und Header

Request parameters

visualFeatures (optional)	string	A string indicating what visual feature types to return. Multiple values should be comma-separated. Valid visual feature types include: <ul style="list-style-type: none">• Categories - categorizes image content according to a taxonomy defined in documentation.• Tags - tags the image with a detailed list of words related to the image content.• Description - describes the image content with a complete English sentence.• Faces - detects if faces are present. If present, generate coordinates, gender and age.• ImageType - detects if image is clipart or a line drawing.• Color - determines the accent color, dominant color, and whether an image is black&white.• Adult - detects if the image is pornographic in nature (depicts nudity or a sex act). Sexually suggestive content is also detected.
details (optional)	string	A string indicating which domain-specific details to return. Multiple values should be comma-separated. Valid visual feature types include: <ul style="list-style-type: none">• Celebrities - identifies celebrities if detected in the image.• Landmarks - identifies landmarks if detected in the image.
language (optional)	string	A string indicating which language to return. The service will return recognition results in specified language. If this parameter is not specified, the default value is "en". Supported languages: <ul style="list-style-type: none">• en - English, Default.• zh - Simplified Chinese.

Request headers

Content-Type (optional)	string	Media type of the body sent to the API.
Ocp-Apim-Subscription-Key	string	Subscription key which provides access to this API. Found in your Cognitive Services accounts .

Beispiel-Code

Code samples

```
Curl  C#  Java  JavaScript  ObjC  PHP  Python  Ruby

using (var content = new ByteArrayContent(byteData))
{
    content.Headers.ContentType = new MediaTypeHeaderValue("< your content type, i.e. application/json >");
    response = await client.PostAsync(uri, content);
}

}
```




9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

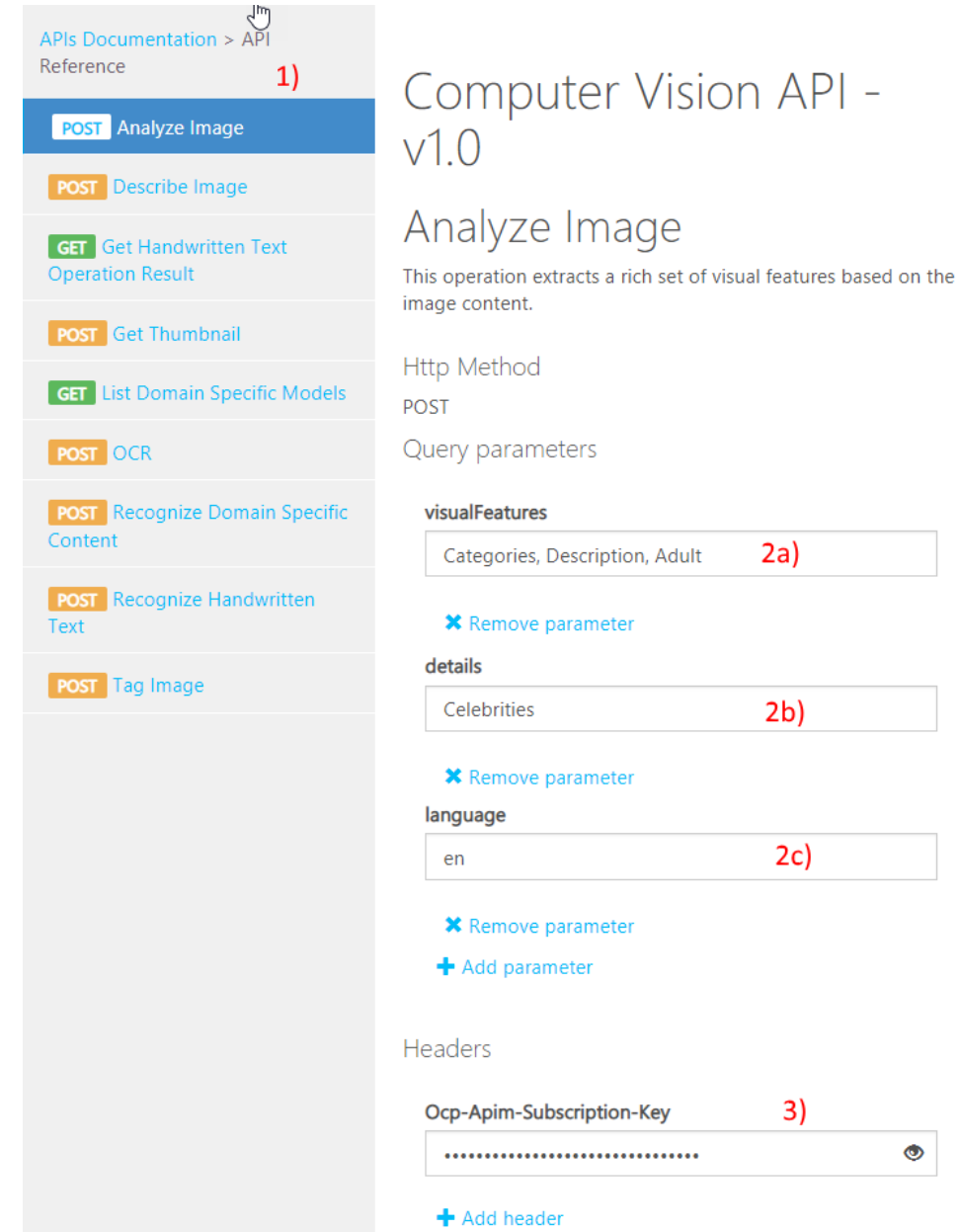
API direkt testen (4)

REST-API testen – Teil 1.



Anweisungen (s. Bild):

1. Jeweilige API auswählen (z.B. „Analyze Image“)
2. Abfrage-Parameter eingeben (s. Letzte Folie N° 24 und API-Referenz Seite – Link „Computer Vision API reference“ beim Schnellstart – s. Folie N° 16 für Details)
3. API-Key im „Ocp-Apim-Subscription-Key“ eingeben (s. P. (1)-(2) auf der Folie N° 22 für Details)
4. Die Seite nach unten scrollen und den Hinweisen auf der nächsten Folie N° 26 folgen.



APIs Documentation > API Reference **1)**

- POST** Analyze Image
- POST** Describe Image
- GET** Get Handwritten Text Operation Result
- POST** Get Thumbnail
- GET** List Domain Specific Models
- POST** OCR
- POST** Recognize Domain Specific Content
- POST** Recognize Handwritten Text
- POST** Tag Image

Computer Vision API - v1.0

Analyze Image

This operation extracts a rich set of visual features based on the image content.

Http Method
POST

Query parameters

visualFeatures

Categories, Description, Adult **2a)**

[✕ Remove parameter](#)

details

Celebrities **2b)**

[✕ Remove parameter](#)


language

en **2c)**

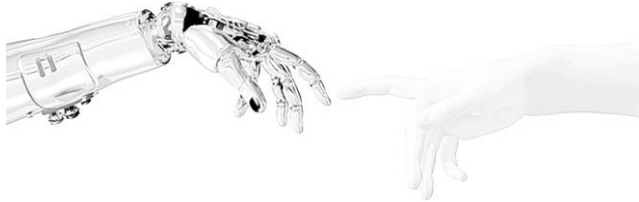
[✕ Remove parameter](#)
[+ Add parameter](#)

Headers

Ocp-Apim-Subscription-Key **3)**

..... 

[+ Add header](#)



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

API direkt testen (5)

REST-API testen – Teil 2.



Anweisungen:

1. Im Feld „Request body“ (s. Bild) eine Bild-Webadresse für „url“-Eigenschaft des JSON-Objektes eintragen – z.B. <https://cdn-s3.si.com/s3fs-public/2013/07/kate-upton10.jpg>
2. Auf „Send“-Button drücken.
3. Hinweisen auf der nächsten Folie N° 27 folgen.

Request body

Input passed within the POST body. Supported input methods: raw image binary or image URL.

Input requirements:

- Supported image formats: JPEG, PNG, GIF, BMP.
- Image file size must be less than 4MB.
- Image dimensions must be at least 50 x 50.

```
1 {"url": "https://cdn-s3.si.com/s3fs-public/2013/07/kate-upton10.jpg"}
```

Request URL

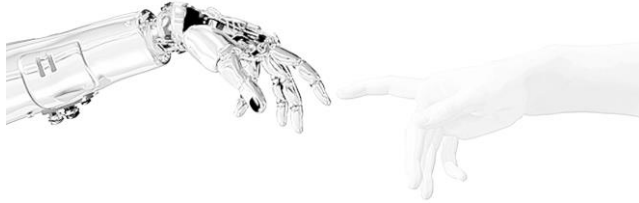
```
https://eastus.api.cognitive.microsoft.com/vision/v1.0/analyze?visualFeatures=Categories, Description, Adult&details=Celebrities&language=en
```

HTTP request

```
POST https://eastus.api.cognitive.microsoft.com/vision/v1.0/analyze?visualFeatures=Categories, Description, Adult&details=Celebrities&language=en HTTP/1.1
Host: eastus.api.cognitive.microsoft.com
Ocp-Apim-Subscription-Key: .....

{"url": "https://cdn-s3.si.com/s3fs-public/2013/07/kate-upton10.jpg"}
```

Send



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

API direkt testen (6)

REST-API testen – Teil 3.

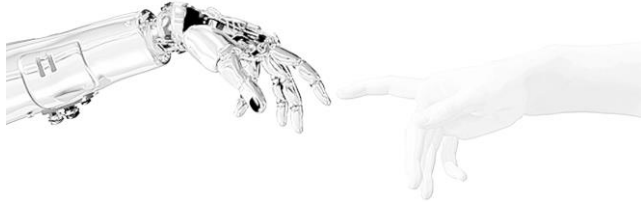


Anweisungen:

1. Dienst-Antwort in JSON-Format von Feld „Respons content“ (s. Bild unten) analysieren.
2. Direkte Test von REST-API für andere kognitive Dienste wiederholen – z.B. für „Describe image“, OCR usw. – s. Folien N°N° 23 bis 27.

Dienst-Antwort in JSON-Format

```
{
  "categories": [{
    "name": "people_",
    "score": 0.99609375,
    "detail": {
      "celebrities": [{
        "name": "Kate Upton",
        "confidence": 0.99998366832733154,
        "faceRectangle": {
          "left": 219,
          "top": 146,
          "width": 65,
          "height": 65
        }
      }]
    }
  }],
  "adult": {
    "isAdultContent": false,
    "isRacyContent": true,
    "adultScore": 0.00067039247369393706,
    "racyScore": 0.997332751750946
  },
  "description": {
    "tags": ["woman", "outdoor", "person", "clothing", "snow", "swimsuit", "beautiful", "girl", "water", "standing", "lady", "female", "wearing", "young", "beach", "surfing", "board", "posing", "white", "smiling", "walking", "black", "suit", "yellow", "holding"],
    "captions": [{
      "text": "Kate Upton wearing a swimsuit",
      "confidence": 0.96940357160609
    }]
  },
  "requestId": "700fe2f6-0516-4024-8240-5719e72665d3",
  "metadata": {
    "width": 467,
    "height": 700,
    "format": "Jpeg"
  }
}
```



9. Demo Lab 1.2

Azure Cognitive Services



Maschinelles Sehen

Bildanalyse

.NET Demo-Projekt

C#-Client mit REST-API



Voraussetzungen

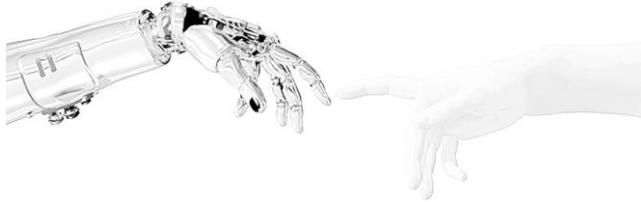
- Eine beliebige Edition von [Visual Studio 2015 oder 2017](#).
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie mithilfe der Visual C#-Konsolen-App-Vorlage eine neue Visual Studio-Projektmappe in Visual Studio.
2. Installieren Sie das NuGet-Paket „Newtonsoft.Json“.
 1. Klicken Sie im Menü auf **Werkzeuge**, wählen Sie **NuGet-Paket-Manager** und dann **NuGet-Pakete für Projektmappe verwalten** aus.
 2. Klicken Sie auf die Registerkarte **Durchsuchen**, und geben Sie „Newtonsoft.Json“ in das Feld **Suchen** ein.
 3. Wählen Sie **Newtonsoft.Json** aus, wenn das Paket angezeigt wird, aktivieren Sie das Kontrollkästchen neben dem Projektnamen, und klicken Sie dann auf **Installieren**.
3. Ersetzen Sie den Code in *Program.cs* durch den [folgenden Code](#)
4. Nehmen Sie dann die folgenden Änderungen im Code vor, falls dies erforderlich ist:
 1. Ersetzen Sie den *subscriptionKey*-Wert durch Ihren Abonnementschlüssel.
 2. Ändern Sie *urlAzureRegion*-Variable bei Bedarf in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
5. Führen Sie das Programm aus.
6. Geben Sie an der Eingabeaufforderung den Pfad zu einem lokalen Bild oder Webadresse von einem Bild im Internet.
7. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **PictureAnalyzer** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services



Maschinelles Sehen

Bildanalyse

.NET Demo-Projekt

C#-Client mit .NET SDK



Voraussetzungen

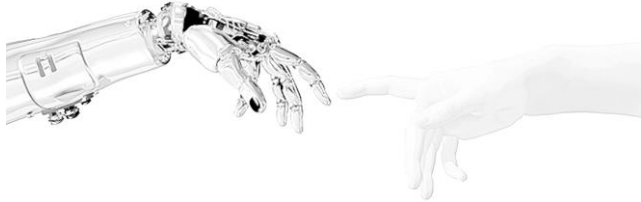
- Eine beliebige Edition von [Visual Studio 2015 oder 2017](#).
- Das NuGet-Paket [Microsoft.Azure.CognitiveServices.Vision.ComputerVision](#) mit der Clientbibliothek. Es ist nicht erforderlich, das Paket herunterzuladen. Unten ist die Installationsanleitung angegeben.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie in Visual Studio eine neue Visual C#-Konsolen-App.
2. Installieren Sie das NuGet-Paket mit der Clientbibliothek der Maschinelles Sehen-API.
 1. Klicken Sie im Menü auf **Werkzeuge**, wählen Sie **NuGet-Paket-Manager** und dann **NuGet-Pakete für Projektmappe verwalten** aus.
 2. Klicken Sie auf die Registerkarte **Durchsuchen**, und geben Sie „Microsoft.Azure.CognitiveServices.Vision.ComputerVision“ in das Feld **Suchen** ein.
 3. Wählen Sie **Microsoft.Azure.CognitiveServices.Vision.ComputerVision** aus, wenn das Paket angezeigt wird, aktivieren Sie das Kontrollkästchen neben dem Projektnamen, und klicken Sie dann auf **Installieren**.
3. Ersetzen Sie *Program.cs* durch den [folgenden Code](#).
4. Ersetzen Sie *subscriptionKey*-Wert durch Ihren gültigen Abonnementschlüssel.
5. Ändern Sie *urlAzureRegion*-Variable bei Bedarf in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
6. Führen Sie das Programm aus.
7. Geben Sie an der Eingabeaufforderung den Pfad zu einem lokalen Bild oder Webadresse von einem Bild im Internet.
8. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **PictureAnalyzerSdk** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services



Maschinelles Sehen

Bildanalyse

Python Demo-Projekt

Python-Client mit REST-API



Voraussetzungen

- Wenn Sie das Beispiel lokal ausführen möchten, muss [Python](#) installiert sein.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Kopieren Sie den [folgenden Code](#) in einen Text-Editor oder in ein Python-IDE wie [PyCharm](#), [Anaconda/Spyder](#), [Visual Studio](#) oder [VS Code](#) etc.
2. Speichern Sie den Code als Datei mit der Erweiterung .py. Beispiel: *PictureAnalyzerPy.py*.
3. Nehmen Sie bei Bedarf die folgenden Änderungen im Code vor:
 1. Ersetzen Sie Wert der *subscription_key*-Variable durch Ihren Abonnementschlüssel.
 2. Ersetzen Sie den Wert von *urlAzureRegion*-Variable durch die in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
 3. Speichern Sie Ihre Änderungen in der Datei *PictureAnalyzerPy.py* ab.
4. Führen Sie das Programm aus – entweder mit Kommandozeile *python PictureAnalyzer.py* oder direkt von jeweiligem Python-IDE aus.
5. Geben Sie an der Eingabeaufforderung den Pfad zu einem lokalen Bild oder Webadresse von einem Bild im Internet.
6. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **PictureAnalyzerPy** öffnen. Dann können Sie die Schritte 1. bis 2. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

Bildanalyse

Java Demo-Projekt

Java-Client mit REST-API



Voraussetzungen

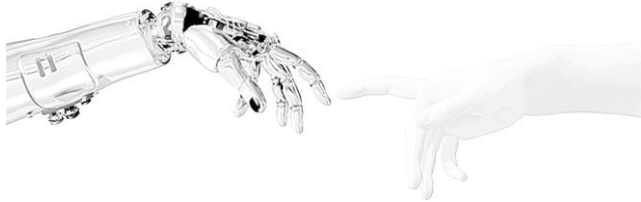
- Die [Java™-Plattform, Standard Edition Development Kit 7 oder 8](#) (JDK 7 oder 8), muss installiert sein. Ein Java IDE wie z.B. [Eclipse](#) ist auch empfohlen.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie in Ihrer bevorzugten Java-IDE oder in Ihrem bevorzugten Editor ein neues Java-Projekt. Wenn die Option verfügbar ist, erstellen Sie das Java-Projekt anhand einer Befehlszeilen-Anwendungsvorlage.
2. Importieren Sie die folgenden Bibliotheken in Ihr Java-Projekt. Wenn Sie Maven verwenden, werden die Maven-Koordinaten für jede Bibliothek bereitgestellt:
 1. [Apache-HTTP-Client](#) (org.apache.httpcomponents:httpclient:4.5.5)
 2. [Apache-HTTP-Core](#) (org.apache.httpcomponents:httpcore:4.4.9)
 3. [JSON-Bibliothek](#) (org.json:json:20180130)
3. Fügen Sie der Datei, die die öffentliche Klasse *Main* für Ihr Projekt enthält, [folgenden Code](#) hinzu.
4. Nehmen Sie bei Bedarf die folgenden Änderungen im Code vor:
 1. Ersetzen Sie den "<API-Key>"-Wert durch Ihren Abonnementschlüssel.
 2. Ersetzen Sie den Wert von *urlAzureRegion*-Variable durch die in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
 3. Weisen Sie der Variable *imageToAnalyze* die Webadresse von einem Bild im Internet zu.
5. Führen Sie das Programm aus – entweder mit Kommandozeile *java Main* oder direkt von jeweiligem Java-IDE aus.
6. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und z.B. Eclipse IDE mit Workspace im geklonten Hauptverzeichnis starten, dann im Projekt *PictureAnalyzerJava* die Datei *Main.java* öffnen. Damit können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services

 Maschinelles Sehen

OCR – Zeichen-Erkennung

.NET Demo-Projekt

C#-Client mit REST-API



Voraussetzungen

- Eine beliebige Edition von [Visual Studio 2015 oder 2017](#).
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie mithilfe der Visual C#-Konsolen-App-Vorlage eine neue Visual Studio-Projektmappe in Visual Studio.
2. Installieren Sie das NuGet-Paket „Newtonsoft.Json“.
 1. Klicken Sie im Menü auf **Werkzeuge**, wählen Sie **NuGet-Paket-Manager** und dann **NuGet-Pakete für Projektmappe verwalten** aus.
 2. Klicken Sie auf die Registerkarte **Durchsuchen**, und geben Sie „Newtonsoft.Json“ in das Feld **Suchen** ein.
 3. Wählen Sie **Newtonsoft.Json** aus, wenn das Paket angezeigt wird, aktivieren Sie das Kontrollkästchen neben dem Projektnamen, und klicken Sie dann auf **Installieren**.
3. Ersetzen Sie den Code in *Program.cs* durch den [folgenden Code](#)
4. Nehmen Sie dann die folgenden Änderungen im Code vor, falls dies erforderlich ist:
 1. Ersetzen Sie den *subscriptionKey*-Wert durch Ihren Abonnementschlüssel.
 2. Ändern Sie *urlAzureRegion*-Variable bei Bedarf in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
5. Führen Sie das Programm aus.
6. Geben Sie an der Eingabeaufforderung den Pfad zu einem lokalen Bild oder Webadresse von einem Bild im Internet, wo ein Text extrahiert werden soll.
7. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **CharacterRecognizerOCR** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services

Maschinelles Sehen

OCR – Zeichen-Erkennung

.NET Demo-Projekt

C#-Client mit .NET SDK



Voraussetzungen

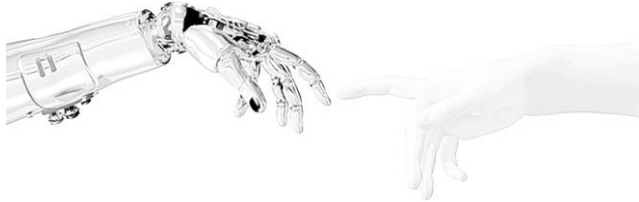
- Eine beliebige Edition von [Visual Studio 2015 oder 2017](#).
- Das NuGet-Paket [Microsoft.Azure.CognitiveServices.Vision.ComputerVision](#) mit der Clientbibliothek. Es ist nicht erforderlich, das Paket herunterzuladen. Unten ist die Installationsanleitung angegeben.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie in Visual Studio eine neue Visual C#-Konsolen-App.
2. Installieren Sie das NuGet-Paket mit der Clientbibliothek der Maschinelles Sehen-API.
 1. Klicken Sie im Menü auf **Werkzeuge**, wählen Sie **NuGet-Paket-Manager** und dann **NuGet-Pakete für Projektmappe verwalten** aus.
 2. Klicken Sie auf die Registerkarte **Durchsuchen**, und geben Sie „Microsoft.Azure.CognitiveServices.Vision.ComputerVision“ in das Feld **Suchen** ein.
 3. Wählen Sie **Microsoft.Azure.CognitiveServices.Vision.ComputerVision** aus, wenn das Paket angezeigt wird, aktivieren Sie das Kontrollkästchen neben dem Projektnamen, und klicken Sie dann auf **Installieren**.
3. Ersetzen Sie *Program.cs* durch den [folgenden Code](#).
4. Ersetzen Sie *subscriptionKey*-Wert durch Ihren gültigen Abonnementschlüssel.
5. Ändern Sie *urlAzureRegion*-Variable bei Bedarf in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
6. Führen Sie das Programm aus.
7. Geben Sie an der Eingabeaufforderung den Pfad zu einem lokalen Bild und eine Webadresse von einem Bild im Internet, wo ein Text extrahiert werden soll.
8. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **CharacterRecognizerOcrSdk** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services

☞ Sprache

Textanalysen

.NET Demo-Projekt

C#-Client mit .NET SDK



Voraussetzungen

- Eine beliebige Edition von [Visual Studio 2015 oder 2017](#).
- Das NuGet-Paket [Microsoft.Azure.CognitiveServices.Language.TextAnalytics](#) mit der Clientbibliothek. Es ist nicht erforderlich, das Paket herunterzuladen. Unten ist die Installationsanleitung angegeben.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Erstellen Sie in Visual Studio eine neue Visual C#-Konsolen-App.
2. Installieren Sie das NuGet-Paket mit der Clientbibliothek der Maschinelles Sehen-API.
 1. Klicken Sie im Menü auf **Werkzeuge**, wählen Sie **NuGet-Paket-Manager** und dann **NuGet-Pakete für Projektmappe verwalten** aus.
 2. Klicken Sie auf die Registerkarte **Durchsuchen**, und geben Sie „Microsoft.Azure.CognitiveServices.Language.TextAnalytics“ in das Feld **Suchen** ein.
 3. Wählen Sie **Microsoft.Azure.CognitiveServices.Language.TextAnalytics** aus, wenn das Paket angezeigt wird, aktivieren Sie das Kontrollkästchen neben dem Projektnamen, und klicken Sie dann auf **Installieren**.
3. Ersetzen Sie *Program.cs* durch den [folgenden Code](#).
4. Ersetzen Sie *subscriptionKey*-Wert durch Ihren gültigen Abonnementschlüssel.
5. Ändern Sie *urlAzureRegion*-Variable bei Bedarf in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
6. Ändern Sie bei Bedarf analysierten Text in den vier Methoden *RunExperiment1()* bis *RunExperiment4()*.
7. Führen Sie das Programm aus.
8. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **TextAnalyzerSdk** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.



9. Demo Lab 1.2

Azure Cognitive Services

🔗 Sprache

Textanalysen

Python Demo-Projekt

Python-Client mit REST-API



Voraussetzungen

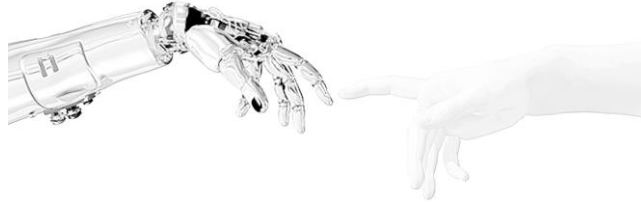
- Wenn Sie das Beispiel lokal ausführen möchten, muss [Python](#) installiert sein.
- Sie benötigen einen Abonnementschlüssel für maschinelles Sehen. Informationen zum Beziehen eines Abonnementschlüssels finden Sie auf Folien N° 20, 22, 39 oder unter [Gewusst wie: Beziehen von Abonnementschlüsseln](#).

Erstellen und Ausführen der Beispielanwendung

Führen Sie die folgenden Schritte durch, um das Beispiel in Visual Studio zu erstellen:

1. Kopieren Sie den [folgenden Code](#) mit Experimenten-Funktionen in einen Text- oder eine Python-IDE Editor wie [PyCharm](#), [Anaconda/Spyder](#), [Visual Studio oder VS Code](#) etc.
2. Speichern Sie den Code mit Experimenten-Funktionen als *TextAnalyzerEx.py*.
3. Kopieren Sie den [folgenden Code](#) des Hauptprogramms in einen Text-Editor oder in Python-IDE.
4. Nehmen Sie bei Bedarf die folgenden Änderungen im Code vor:
 1. Ersetzen Sie Wert der *subscription_key*-Variable durch Ihren Abonnementschlüssel.
 2. Ersetzen Sie den Wert von *urlAzureRegion*-Variable durch die in die Azure-Region, die Ihren Abonnementschlüsseln zugeordnet ist, wenn erforderlich.
5. Speichern Sie den Hauptprogramm-Code ab – z.B. als *TextAnalyzerPy.py*.
6. In der Datei *TextAnalyzerEx.py* ändern Sie bei Bedarf analysierten Text in den vier Methoden *Experiment_1()* bis *Experiment_4()* und speichern Sie anschließend Ihre Änderungen ab.
7. Führen Sie das Programm *TextAnalyzerPy.py* aus – entweder mit Kommandozeile *python PictureAnalyzer.py* oder direkt von jeweiligem Python-IDE.
8. Analysieren Sie die JSON-Result.

Achtung! Sie können die gesamte [Visual Studio Projektmappe für Lab 1-2 von GitHub](#) klonen und das Projekt **PictureAnalyzerPy** öffnen. Dann können Sie die Schritte 1. bis 3. überspringen.

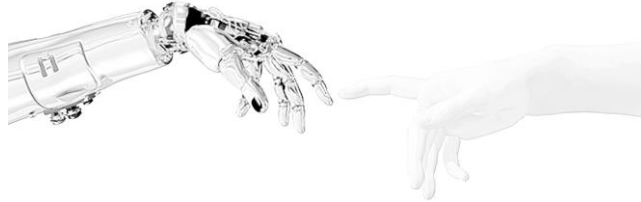


9. Demo Lab 1.2

Azure Cognitive Services

Weitere Schritte

1. Weitere kognitive Dienste (s. Folien N° 8. bis 13) über Web-Portal exemplarisch erkunden, evaluieren und testen so wie es für Bildanalyse-Dienst (s. Folien N° 14 bis 27) gemacht worden ist.
2. Für ausgewählte kognitive Dienste einen Dienst-API Client in einer passenden Programmiersprache erstellen so wie es z.B. für Bildanalyse-Dienst (s. Folien N° 28 bis 33) und Textanalyse-Dienst (s. Folien N° 34,35) gemacht worden ist.
3. Ressourcen und Links (nächste Folie N° 37) nutzen.
4. Firma Software Factories kann bei der Umsetzung mit Azure Cognitive eines bestimmten Szenarios und Anwendungsfalles ihre Kunden unterstützen bzw. eine Lösung für das gegebene Problem anbieten (s. Kontaktdaten auf der letzten Folie)

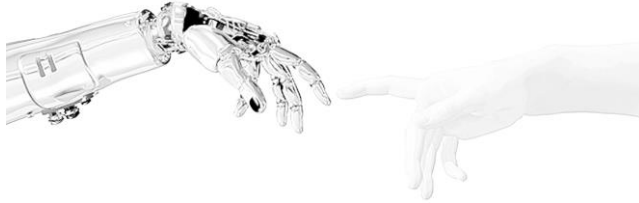


9. Demo Lab 1.2

Azure Cognitive Services

Ressourcen & Links.

1. [Materialien zum aktuellen Lab 1.2](#)
2. [Labs-Projekte zum Kurs "KI, Deep und Machine Learning"](#)
3. [Azure Cognitive Services ausprobieren \(API-Schlüssel\)](#)
4. [Azure Machine Learning. Dokumentation](#)
5. [Azure Cognitive Services. Dokumentation](#)
6. [Azure Cognitive Services. Hauptseite](#)
7. [Azure Cognitive Services. Verzeichnis](#)
8. [Azure Cognitive Services. Bildanalyse](#)
9. [Azure Cognitive Services. Spracheingabe](#)
10. [Azure Cognitive Services. Sprache](#)
11. [Azure Cognitive Services. Einblicke und Wissen](#)
12. [Azure Cognitive Services. Suchen](#)
13. [M. Dobrochynskyy. Gesicht, Emotionen, Videos. dotnetpro 2017/12](#)
14. [M. Dobrochynskyy. Moderieren mit Automatik. dotnetpro 2017/09](#)
15. [M. Dobrochynskyy. Spracheingabe. dotnetpro 2018/02](#)



9. Demo Lab 1.2

Azure Cognitive Services



Danke!

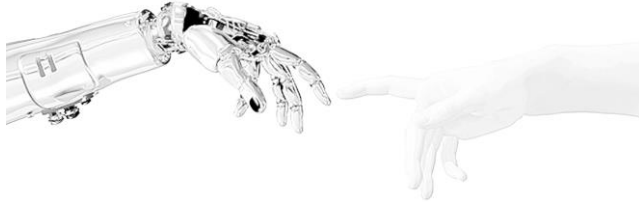
Mykola Dobrochynskyy ist Geschäftsführer von Software Factories. Sein Fokus und seine Interessen sind Modellgetriebene Softwareentwicklung, Code Generierung, künstliche Intelligenz, Machine und Deep Learning sowie Cloud und Service orientierte Software-Architekturen.

@my_dobro

ceo@soft-fact.de

<http://www.software-factories.de>

<http://new.machine-learning.events>



9. Demo Lab 1.2 Azure Cognitive Services

Bonus-Info. API-Keys von Cognitive Services in Visual Studio

1. Im Server-Explorer Fenster von Visual Studio 2017 „AI Tools“ öffnen (wenn noch nicht vorhanden, via Menü „Extras – Extensions & Updates“ installieren).
2. Relevanten Dienst auswählen (s. 2) auf dem Bild).
3. Im Kontext-Menü aufrufen (z.B. via rechte Maus-Taste).
4. „Subscriptions Key“ anklicken.
5. Key1 oder Key2 in die Zwischenablage kopieren (s. 5) auf dem Bild unten).

