

Caffe vom Berkeley Vision and Learning Center (BVLC)

Kategorien : [Software](#), [Open Source](#)

Schlagwörter : [Deep Learning](#), [Künstliche Intelligenz](#), [Maschinelles Lernen](#)

Datum : 31. Mai 2017

LABELS:



| | |
|-------------------|-----|
| Hair | 95% |
| Clothing | 92% |
| Beauty | 82% |
| Hairstyle | 78% |
| Fashion | 75% |
| Head | 74% |
| Black Hair | 67% |
| Model | 66% |
| Long Hair | 66% |
| Fashion Accessory | 63% |
| Photo Shoot | 56% |

Die wahre Herausforderung an die [Künstliche Intelligenz](#) besteht in der Lösung von Aufgaben, die für uns Menschen leicht durchzuführen sind, aber deren Lösung nur schwer als mathematische Regeln zu formulieren sind. Dies sind Aufgaben, die wir Menschen intuitiv lösen, wie z.B. [Spracherkennung](#) oder [Bilderkennung](#).

Eine computerbasierte Lösung für diese Art von Aufgaben beinhaltet die Fähigkeit von Computern, aus der Erfahrung zu lernen und die Welt in Bezug auf eine Hierarchie von Konzepten zu verstehen. Hierbei ist jedes Konzept durch seine Beziehung zu einfacheren Konzepten definiert. Durch das Sammeln von Wissen aus der Erfahrung vermeidet dieser Ansatz die Notwendigkeit für die menschlichen Bediener, all das Wissen, das der Computer für seine Arbeit benötigt, formal spezifizieren zu müssen. Die Hierarchie der Konzepte erlaubt es dem Computer komplizierte Konzepte zu erlernen, indem er sie aus einfacheren zusammensetzt. Wenn wir ein Diagramm zeichnen, das zeigt, wie diese Konzepte übereinander aufgebaut werden, dann ist das Diagramm

tief, mit vielen Schichten. Aus diesem Grund nennen wir diesen Ansatz in der Künstlichen Intelligenz „Deep Learning“.

Caffe ist eine Programmbibliothek für Deep Learning.

Sie wurde von Yangqing Jia während seiner Ph.D.-Zeit am Vision und Learning Centre der University of California, Berkeley entwickelt.

Caffe hat zuerst die MATLAB-Implementierung von schnellen Convolutional Neural Networks (CNN) nach C und C++ portiert. Caffe beinhaltet zahlreiche Algorithmen und Deep-Learning-Architekturen für die Klassifikation und Clusteranalyse von Bilddaten. CNN, R-CNN (Rekurrentes neuronales Netz), LSTM (Long short-term memory) und vollständig verbundene neuronale Netze werden unterstützt. Mit Caffe kann die Grafikprozessor-basierte Beschleunigung mit cuDNN von Nvidia genutzt werden, **sodass 60 Millionen Bilder pro Tag prozessiert werden können.**

Als hauptsächliche Programmierschnittstelle sind Python (NumPy) und MATLAB vorgesehen. Yahoo hat Caffe in Apache Spark eingebunden ([CaffeOnSpark](#)) um Deep Learning verteilt zu verwenden.

Eine der häufigsten Techniken in der Künstlichen Intelligenz ist Maschinelles Lernen. Maschinelles Lernen ist ein selbstadaptiver Algorithmus. Deep Learning, eine Teilmenge des Maschinellen Lernens, nutzt eine Reihe hierarchischer Schichten bzw. eine Hierarchie von Konzepten um den Prozess des Maschinellen Lernens durchzuführen. Die hierbei benutzten Künstlichen neuronalen Netze sind wie das menschliche Gehirn gebaut, wobei die Neuronen wie ein Netz miteinander verbunden sind. Die erste Schicht des neuronalen Netzes, der sichtbare „input Layer“, verarbeitet eine Rohdateneingabe, wie z.B. die einzelnen Pixel eines Bildes. Die Dateneingabe enthält Variablen, die wir beobachten können, daher „sichtbare Schicht“.

Windows Caffe

LABELS:



| | |
|-------------------|-----|
| Hair | 95% |
| Clothing | 92% |
| Beauty | 82% |
| Hairstyle | 78% |
| Fashion | 75% |
| Head | 74% |
| Black Hair | 67% |
| Model | 66% |
| Long Hair | 66% |
| Fashion Accessory | 63% |
| Photo Shoot | 56% |

Die Entwickler betreiben Caffe auf Ubuntu 16.04-12.04, OS X 10.11-10.8 und über Docker und AWS. Die Makefile.config und Makefile.config Build werden durch eine Community-CMake-Build ergänzt. Guillaume Dumont pflegt einen Windows branch auf GitHub. Da unsere Bilddateien in der Regel auf Windows Festplatten liegen, freuen wir uns besonders über diesen Supprt. Ihr findet Windows Caffe hier: <https://github.com/BVLC/caffe/tree/windows>

```
caffe.exe: command line brew usage: caffe          commands:      train
      train or finetune a model      test          score a model  device_
query      show GPU diagnostic information  time          benchmark mod
el execution time          Flags from C:\Users\guillaume\work\caffe-builder\b
uild_v140_x64\packages\gflags\gflags_download-prefix\src\gflags_download\
src\gflags.cc:      -flagfile (load flags from file) type: string default
: ""      -fromenv (set flags from the environment [use 'export FLAGS fla
gl=value'])          type: string default: ""      -tryfromenv (set flags f
rom the environment if present) type: string          default: ""      -und
```

```
efok (comma-separated list of flag names that it is okay to specify on
the command line even if the program does not define a flag with tha
t name. IMPORTANT: flags in this list that have arguments MUST use the
flag=value format) type: string default: "" Flags from
C:\Users\guillaume\work\caffe-builder\build_v140_x64\packages\gflags\gfla
gs_download-prefix\src\gflags_download\src\gflags_completions.cc: -t
ab_completion_columns (Number of columns to use in output for tab
completion) type: int32 default: 80 -tab_completion_word (If non-emp
ty, HandleCommandLineCompletions() will hijack the process and att
empt to do bash-style command line flag completion on this value.)
type: string default: "" Flags from C:\Users\guillaume\work\caffe-b
uilder\build_v140_x64\packages\gflags\gflags_download-prefix\src\gflags_d
ownload\src\gflags_reporting.cc: -help (show help on all flags [tip:
all flags can have two dashes]) type: bool default: false current
ly: true -helpfull (show help on all flags -- same as -help) type: b
ool default: false -helpmatch (show help on modules whose nam
e contains the specified substr) type: string default: "" -he
lpon (show help on the modules named by this flag value) type: string
default: "" -helppackage (show help on all modules in the main p
ackage) type: bool default: false -helpshort (show help on on
ly the main module for this program) type: bool default: false
-helpxml (produce an xml version of help) type: bool default: false
-version (show version and build info and exit) type: bool default: fal
se Flags from C:\Users\guillaume\work\caffe-builder\build_v140_x
64\packages\glog\glog_download-prefix\src\glog_download\src\logging.cc:
-alsologtoemail (log messages go to these email addresses in addition
to logfiles) type: string default: "" -alsologtostderr (log
messages go to stderr in addition to logfiles) type: bool default:
false currently: true -colorlogtostderr (color messages logged to s
tderr (if supported by terminal)) type: bool default: false -
log_backtrace_at (Emit a backtrace when logging at file:linenum.)
type: string default: "" -log_dir (If specified, logfiles are writte
n into this directory instead of the default logging directory.) t
ype: string default: "" -log_link (Put additional links to the log f
iles in this directory) type: string default: "" -log_prefix
(Prepend the log prefix to the start of each log line) type: bool
default: true -logbuflevel (Buffer log messages logged at this level
or lower (-1 means don't buffer; 0 means buffer INFO only; ...))
type: int32 default: 0 -logbufsecs (Buffer log messages for at most
this many seconds) type: int32 default: 30 -logemaillevel (Em
ail log messages logged at this level or higher (0 means email all
; 3 means email FATAL only; ...)) type: int32 default: 999 -logfile_
```

```
mode (Log file mode/permissions.) type: int32 default: 436      -logmaile
r (Mailer used to send logging email) type: string      default: "/bin/
mail"      -logtostderr (log messages go to stderr instead of logfiles) t
ype: bool      default: false      -max_log_size (approx. maximum log f
ile size (in MB). A value of 0 will be      silently overridden to 1.)
type: int32 default: 1800      -minloglevel (Messages logged at a lower l
evel than this don't actually get      logged anywhere) type: int32 def
ault: 0      -stderrthreshold (log messages at or above this level are co
pied to stderr      in addition to logfiles. This flag obsoletes --als
ologtostderr.)      type: int32 default: 2      -stop_logging_if_full_d
isk (Stop attempting to log to disk if the disk is      full.) type: bo
ol default: false      Flags from C:\Users\guillaume\work\caffe-builder\b
uild_v140_x64\packages\glog\glog_download-prefix\src\glog_download\src\vl
og_is_on.cc:      -v (Show all VLOG(m) messages for m <= this. Overridabl
e by --vmodule.)      type: int32 default: 0      -vmodule (per-module
verbose level. Argument is a comma-separated list of      =. is a glob
pattern, matched      against the filename base (that is, name ignorin
g .cc/.h./-inl.h). overrides any value given by --v.) type: string defau
lt: ""      Flags from C:\projects\caffe\tools\caffe.cpp:      -gpu (
Optional; run in GPU mode on given device IDs separated by ','.Use
'-gpu all' to run on all available GPUs. The effective training batch
size is multiplied by the number of devices.) type: string default:
""      -iterations (The number of iterations to run.) type: int32 defaul
t: 50      -level (Optional; network level.) type: int32 default: 0
-model (The model definition protocol buffer text file.) type: string
default: ""      -phase (Optional; network phase (TRAIN or TEST). Onl
y used for 'time'.)      type: string default: ""      -sighup_effect (
Optional; action to take when a SIGHUP signal is received:      snapsho
t, stop or none.) type: string default: "snapshot"      -sigint_effect (O
ptional; action to take when a SIGINT signal is received:      snapshot
, stop or none.) type: string default: "stop"      -snapshot (Optional; t
he snapshot solver state to resume training.)      type: string default
: ""      -solver (The solver definition protocol buffer text file.) type
: string      default: ""      -stage (Optional; network stages (not to
be confused with phase), separated      by ','.) type: string default:
""      -weights (Optional; the pretrained weights to initialize finetun
ing,      separated by ','. Cannot be set simultaneously with snapshot.
)      type: string default: ""
```

Bilddatenbank vorbereiten

Zum Testen der Trainingsleistung von Caffe ist als Input eine Bilddatenbank erforderlich. Caffe umfasst Modelle, die gemäß ihrer Konfiguration Bilder aus der ILSVRC12-Challenge („ImageNet“) verwenden. Die Originalbilddateien stehen unter <http://image-net.org/download-images> zum Herunterladen zur Verfügung.

Auf den neuesten NVIDIA Pascal™-Grafikprozessoren läuft Caffe bis zu 65 % schneller. Innerhalb eines einzelnen Knotens lässt es sich über mehrere Grafikprozessoren skalieren. Dadurch lassen sich Modelle innerhalb weniger Stunden anstatt mehrerer Tage trainieren.

Nützliche Links

[Caffe](#)

[Windows Caffe](#)

[Bachelorarbeit: Erkennung von Verkehrszeichen mit Konvolutionalen Neuronalen Netzen](#)

<http://www.nvidia.de/object/caffe-deep-learning-framework-de.html>

Jetzt bist du gefragt!

Bereits heute ermöglicht Deep Learning selbstfahrende Autos, intelligente persönliche Assistenten und intelligentere Webdienste. Hast du Anregungen oder Ergänzungen? Dann freue ich mich auf deinen Kommentar.

Du kannst diesen Beitrag natürlich auch weiterempfehlen. Ich bin dir für jede Unterstützung dankbar!

Verwandeln Sie Ihren Commerce mit AR und 3D-Produktvisualisierung!

Bei uns geht es um Techniken, die es schaffen, das Produkt zum Erlebnis zu machen. Virtual & Augmented Reality, 360 Grad-Videos, Darstellungen in 3D, virtuelle Showrooms. Die Besucher:innen sollen eintauchen in die Welt des Unternehmens mit immersiven Technologien.



Sie können uns mit der Erstellung von individuellen 3D-Visualisierungen beauftragen. Jeder kann 3D-Visualisierungen bei unserem Kreativservice bestellen - unabhängig davon, ob Sie nur ein einzelnes 3D-Modell benötigen oder viele.

Wir unterstützen Sie bei der Umsetzung Ihres Augmented Reality (AR) oder Virtual Reality (VR) Projektes! Egal ob [Produktfotografie](#), [3D Scan Service](#), [3D-Visualisierung](#) oder fertige [3D Modelle für AR/VR](#) – wir beraten Sie persönlich und unverbindlich.

Wo kann ich Anregungen, Lob oder Kritik äußern?

Ihre Meinung ist uns wichtig! Schreiben Sie uns, was Ihnen in Bezug auf unser Angebot bewegt. info@r23.de

R23 — Ihre Digitalagentur für Virtual Reality und interaktive Markenerlebnisse

Wünschen Sie ein individuelles Angebot auf Basis Ihrer aktuellen Vorlagen, nutzen Sie einfach unser [Anfrageformular](#).

Caffe vom Berkeley Vision and Learning Center (BVLC)



<https://blog.r23.de/caffe-vom-berkeley-vision-and-learning-center-bvlc/>

Besuchen Sie uns auch auf [Facebook](#) und [Twitter](#).

r23
Thüringenstr. 20
58135 Hagen
Deutschland
Telefon: 02331 / 9 23 21 29

E-Mail: info@r23.de

Ust-IdNr.:DE250502477