



(10) **DE 10 2024 204 143 A1** 2024.11.14

(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2024 204 143.2**
(22) Anmeldetag: **03.05.2024**
(43) Offenlegungstag: **14.11.2024**

(51) Int Cl.: **G06V 10/70 (2022.01)**
G06T 11/00 (2006.01)

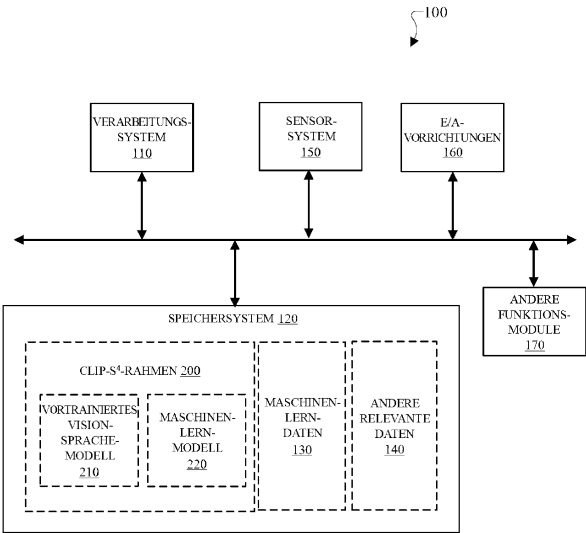
(30) Unionspriorität: 18/316,690 12.05.2023 US	(72) Erfinder: He, Wenbin, Sunnyvale, CA, US; Gou, Liang, San Jose, CA, US; Jamonnak, Suphanut, Santa Clara, CA, US; Ren, Liu, Saratoga, CA, US
(71) Anmelder: Robert Bosch Gesellschaft mit beschränkter Haftung, 70469 Stuttgart, DE	

Prüfungsantrag gemäß § 44 PatG ist gestellt.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen.

(54) Bezeichnung: **System und Verfahren mit sprachengeführter selbstbeaufsichtigter semantischer Segmentierung**

(57) Zusammenfassung: Ein computerimplementiertes System und Verfahren betreffen sprachengeführte selbstbeaufsichtigte semantische Segmentierung. Es wird ein modifiziertes Bild erzeugt durch Ausführen von Datenergänzung an einem Quellenbild. Ein Maschinenlernmodell erzeugt erste Pixeleinbettungen auf der Basis des modifizierten Bildes. Erste Segmenteinbettungen werden unter Verwendung der ersten Pixeleinbettungen erzeugt. Ein vortrainiertes Vision-Sprache-Modell erzeugt zweite Pixeleinbettungen auf der Basis des Quellenbildes. Zweite Segmenteinbettungen werden erzeugt durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird. Es werden einbettungseinheitliche Verlustdaten erzeugt durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen. Es werden kombinierte Verlustdaten erzeugt, die die einbettungseinheitlichen Verlustdaten umfassen. Parameter des Maschinenlernmodells werden auf der Basis der kombinierten Verlustdaten aktualisiert.



Beschreibung**Gebiet**

[0001] Die vorliegende Offenbarung betrifft allgemein Computervision und insbesondere digitale Bildverarbeitung, visuelle Analytik und semantische Segmentierung.

Hintergrund

[0002] Im Allgemeinen umfasst semantische Segmentierung Aufteilen eines Eingangsbildes in semantisch bedeutsame Regionen auf Pixelebene und Versehen jeder Region mit einer semantischen Klassenkennzeichnung. Neuere Fortschritte bei der semantischen Segmentierung stützen sich in hohem Maße auf pixelweise menschliche Annotationen. Die Beschaffung von pixelweisen Annotationen ist jedoch äußerst arbeitsintensiv und kostspielig. Zum Beispiel hat es sich gezeigt, dass pixelweise Annotationen zur Kennzeichnung eines Bildes 1,5 Stunden in Anspruch nehmen. Außerdem sind menschliche Annotationen oft auf vordefinierte semantische Klassen beschränkt, wodurch die Erkennungsfähigkeiten der gelehnten Modelle auf diese vordefinierten semantischen Klassen beschränkt werden können.

Kurzfassung

[0003] Es folgt eine Kurzfassung bestimmter Ausführungsformen, die nachfolgend ausführlich beschrieben werden. Die beschriebenen Aspekte werden lediglich dargestellt, um dem Leser eine kurze Zusammenfassung dieser bestimmten Ausführungsformen bereitzustellen, und die Beschreibung dieser Aspekte soll den Schutzzumfang der vorliegenden Offenbarung nicht begrenzen. Tatsächlich kann diese Offenbarung vielfältige Aspekte einschließen, die nachfolgend möglicherweise nicht ausdrücklich dargelegt werden.

[0004] Gemäß mindestens einem Aspekt betrifft ein computerimplementiertes Verfahren sprachengeführte selbstbeaufsichtigte semantische Segmentierung. Das Verfahren umfasst Empfangen eines Quellenbildes. Das Verfahren umfasst Erzeugen eines modifizierten Bildes durch Ausführen von Datenergänzung an dem Quellenbild. Das modifizierte Bild ist eine modifizierte Version des Quellenbildes. Das Verfahren umfasst Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell. Das Verfahren umfasst Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen. Das Verfahren umfasst Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell. Das Verfahren umfasst Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird. Das Verfahren umfasst Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen. Das Verfahren umfasst Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen. Das Verfahren umfasst Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten. Das Verfahren umfasst Bereitstellen oder Ausgeben eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische

gen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird. Das Verfahren umfasst Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen. Das Verfahren umfasst Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen. Das Verfahren umfasst Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten. Das Verfahren umfasst Bereitstellen eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische Segmentierung. Der Rahmen umfasst (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden.

[0005] Gemäß mindestens einem Aspekt betrifft ein System sprachengeführte selbstbeaufsichtigte semantische Segmentierung. Das System umfasst mindestens einen Prozessor und ein nichtflüchtiges computerlesbares Medium. Das nichtflüchtige computerlesbare Medium befindet sich in Datenkommunikation mit dem Prozessor. Das nichtflüchtige computerlesbare Medium weist computerlesbare Daten mit darauf gespeicherten Anweisungen auf, die, wenn sie durch den Prozessor ausgeführt werden, dafür ausgelegt sind, zu bewirken, dass der Prozessor ein Verfahren ausführt. Das Verfahren umfasst Empfangen eines Quellenbildes. Das Verfahren umfasst Erzeugen eines modifizierten Bildes durch Ausführen von Datenergänzung an dem Quellenbild. Das modifizierte Bild ist eine modifizierte Version des Quellenbildes. Das Verfahren umfasst Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell. Das Verfahren umfasst Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen. Das Verfahren umfasst Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell. Das Verfahren umfasst Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird. Das Verfahren umfasst Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen. Das Verfahren umfasst Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen. Das Verfahren umfasst Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten. Das Verfahren umfasst Bereitstellen oder Ausgeben eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische

Segmentierung. Der Rahmen umfasst (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden.

[0006] Gemäß mindestens einem Aspekt weist ein nichtflüchtiges computerlesbares Medium computerlesbare Daten mit darauf gespeicherten Anweisungen auf. Die computerlesbaren Daten sind durch einen oder mehrere Prozessoren ausführbar, um ein Verfahren auszuführen. Das Verfahren umfasst Empfangen eines Quellenbildes. Das Verfahren umfasst Erzeugen eines modifizierten Bildes durch Ausführen von Datenergänzung an dem Quellenbild. Das modifizierte Bild ist eine modifizierte Version des Quellenbildes. Das Verfahren umfasst Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell. Das Verfahren umfasst Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen. Das Verfahren umfasst Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell. Das Verfahren umfasst Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird. Das Verfahren umfasst Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen. Das Verfahren umfasst Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen. Das Verfahren umfasst Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten. Das Verfahren umfasst Bereitstellen eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische Segmentierung. Der Rahmen umfasst (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden.

[0007] Diese und andere Merkmale, Aspekte und Vorteile der vorliegenden Erfindung werden in der folgenden ausführlichen Beschreibung gemäß den beigefügten Zeichnungen besprochen, in denen gleiche Zeichen durchweg ähnliche oder gleiche Teile repräsentieren.

Kurze Beschreibung der Zeichnungen

Fig. 1 ist eine Darstellung eines Beispiels für ein System mit einem Rahmen für sprachengeführte selbstbeaufsichtigte semantische Segmentierung gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 2A ist ein Flussdiagramm, das Aspekte des Trainings des Rahmens von **Fig. 1** gemäß einer

beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 2B ist eine Darstellung eines nicht einschränkenden Beispiels, das Aspekte des Erzeugens von ersten Verlustdaten für den Rahmen von **Fig. 2A** gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 2C ist eine Darstellung eines nicht einschränkenden Beispiels, das Aspekte des Erzeugens von zweiten Verlustdaten für den Rahmen von **Fig. 2A** gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 2D ist eine Darstellung eines nicht einschränkenden Beispiels, das Aspekte des Erzeugens von dritten Verlustdaten für den Rahmen von **Fig. 2A** gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 3 ist eine Darstellung eines nicht einschränkenden Beispiels, das Aspekte des Berechnens von Zielklassenprototypen gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 4A ist eine Darstellung eines nicht einschränkenden Beispiels für einen Einbettungsraum, der Semantik umfasst, worin eine Menge von Pixeleinbettungen aus dem vortrainierten Vision-Sprache-Modell nicht mit einer Menge von Pixeleinbettungen aus dem Maschinenlernmodell ausgerichtet ist, gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 4B ist eine Darstellung eines nicht einschränkenden Beispiels für einen Einbettungsraum, der Semantik umfasst, worin eine Menge von Pixeleinbettungen aus dem vortrainierten Vision-Sprache-Modell mit einer Menge von Pixeleinbettungen aus dem Maschinenlernmodell ausgerichtet ist, gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 5 ist eine Darstellung eines nicht einschränkenden Beispiels, dass verschiedene Aspekte des Rahmens von **Fig. 2A** während Schlussfolgerung gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung veranschaulicht.

Fig. 6 ist eine Darstellung eines Beispiels für ein Steuersystem, das den Rahmen von **Fig. 2A** während Schlussfolgerung verwendet, gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 7 ist eine Darstellung des Steuersystems von **Fig. 6** mit Bezug auf mobile Maschinentech-

nologie gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 8 ist eine Darstellung des Steuersystems von **Fig. 6** mit Bezug auf Sicherheitstechnologie gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Fig. 9 ist eine Darstellung des Steuersystems von **Fig. 6** mit Bezug auf Bildgebungstechnologie gemäß einer beispielhaften Ausführungsform der vorliegenden Offenbarung.

Ausführliche Beschreibung

[0008] Die hier beschriebenen Ausführungsformen, die anhand von Beispielen gezeigt und beschrieben wurden, und viele ihrer Vorteile werden aus der obigen Beschreibung verständlich, und es ist ersichtlich, dass verschiedene Änderungen an Form, Konstruktion und Anordnung der Komponenten vorgenommen werden können, ohne vom offenbarten Gegenstand abzuweichen oder einen oder mehrere seiner Vorteile zu opfern. Tatsächlich sind die beschriebenen Formen dieser Ausführungsformen lediglich erläuternd. Es gibt verschiedene Modifikationen und alternative Formen dieser Ausführungsformen, und die folgenden Ansprüche sollen solche Änderungen einschließen und umfassen und nicht auf die bestimmten Formen beschränkt sein, sondern alle Modifikationen, Äquivalente und Alternativen abdecken, die in den Gedanken und Schutzzumfang der vorliegenden Offenbarung fallen.

[0009] **Fig. 1** ist eine Darstellung eines nicht einschränkenden Beispiels für ein System 100 mit einem Rahmen 200, das für sprachengeführte selbstbeaufsichtigte semantische Segmentierung („CLIP-S4“) ausgelegt ist. Der Rahmen 200 ist ausgelegt zum Lernen von einheitlichen Pixeleinbettungen mit Bezug auf visuelle und konzeptuelle Semantik unter Verwendung von selbstbeaufsichtigtem Lernen und der Führung eines vortrainierten Vision-Sprache-Modells 210, z. B. des kontrastiven Sprache-Bild-Vortrainings (CLIP). Der Rahmen 200 stellt einen Ansatz der selbstbeaufsichtigten semantischen Segmentierung bereit, der pixel-segmentkontrastives Lernen mit der Führung eines vortrainierten Vision-Sprache-Modells 210 kombiniert. Der Rahmen 200 ist ausgelegt zum Erzeugen von qualitativ hochwertigen Pixeleinbettungen ohne jegliche menschliche Annotationen. Der Rahmen 200 ist auf vielfältige Aufgaben der semantischen Segmentierung anwendbar (z. B. sprachengesteuerte semantische Segmentierung für sowohl bekannte als auch unbekannte Klassen, unbeaufsichtigte semantische Segmentierung mit k-Means-Clusterung/linearer Klassifizierung, Transferlernen von erzeugten Pixeleinbettungen zur Instanz-Maske-Verfolgung usw.). Außerdem stellt der Rahmen 200 sprachengesteuerte semantische Segmentierung ohne jegliche

menschliche Annotationen bereit durch Einführen und Behandeln einer neuen Aufgabe der klassenfreien semantischen Segmentierung. Statt anzunehmen, dass alle Klassennamen während des Trainings bekannt sind, ist der Rahmen 200 dahingehend vorteilhaft, dass er dafür ausgelegt ist, unbekannte Klassen aus ungekennzeichneten Bild-daten zu entdecken, ohne unbekannte Klassennamen überhaupt zu kennen.

[0010] Das System 100 umfasst mindestens ein Verarbeitungssystem 110 mit mindestens einer Verarbeitungsvorrichtung. Zum Beispiel umfasst das Verarbeitungssystem 110 mindestens einen elektronischen Prozessor, eine CPU (Zentralverarbeitungseinheit), eine GPU (Grafikverarbeitungseinheit), einen Mikroprozessor, ein FPGA (Field Programmable Gate Array), eine ASIC (anwendungsspezifische integrierte Schaltung), eine beliebige geeignete Verarbeitungstechnologie oder eine beliebige Anzahl und Kombination davon. Das Verarbeitungssystem 110 ist betreibbar, um die hier beschriebene Funktionalität bereitzustellen.

[0011] Das System 100 umfasst ein Speichersystem 120, das wirksam mit dem Verarbeitungssystem 110 verbunden ist. Bei einer beispielhaften Ausführungsform umfasst das Speichersystem 120 mindestens ein nichtflüchtiges computerlesbares Speichermedium, das dafür ausgelegt ist, verschiedene Daten zu speichern und Zugang zu diesen bereitzustellen, um es dem Verarbeitungssystem 110 zu ermöglichen, die Operationen und Funktionalität wie hier offenbart auszuführen. Bei einer beispielhaften Ausführungsform umfasst das Speichersystem 120 eine einzige Speichervorrichtung oder mehrere Speichervorrichtungen. Das Speichersystem 120 kann elektrische, elektronische, magnetische, optische, Halbleiter-, elektromagnetische oder beliebige geeignete Speicherungstechnologie umfassen, die in dem Rahmen 200 betreibbar ist. Zum Beispiel kann bei einer beispielhaften Ausführungsform das Speichersystem 120 RAM (Direktzugriffsspeicher), ROM (Nurlese-speicher), Flash-Speicher, ein Plattenlaufwerk, eine Speicherkarte, eine optische Speichervorrichtung, eine magnetische Speichervorrichtung, ein Speichermodul, eine beliebige Art von Speichervorrichtung oder eine beliebige Anzahl und Kombination davon umfassen. Mit Bezug auf das Verarbeitungssystem 110 und/oder andere Komponenten des Rahmens 200 ist das Speichersystem 120 lokal, entfernt oder eine Kombination davon (z. B. teilweise lokal und teilweise entfernt). Zum Beispiel kann das Speichersystem 120 mindestens ein Speichersystem auf Cloud-Basis (z. B. ein Datenbanksystem auf Cloud-Basis) umfassen, das von dem Verarbeitungssystem 110 und/oder anderen Komponenten des Rahmens 200 entfernt ist.

[0012] Das Speichersystem 120 umfasst mindestens einen Rahmen 200, Maschinenlernenden 130 und andere relevante Daten 140, die darauf gespeichert sind. Speziell umfasst der Rahmen 200 computerlesbare Daten mit Anweisungen, die, wenn sie durch das Verarbeitungssystem 110 ausgeführt werden, dafür ausgelegt sind, über den Rahmen 200 sprachengeführte selbstbeaufsichtigte semantische Segmentierung zu implementieren. Die computerlesbaren Daten können Anweisungen, Code, Routinen, verschiedene diesbezügliche Daten, eine beliebige Softwaretechnologie oder eine beliebige Anzahl und Kombination davon umfassen. Der Rahmen 200 umfasst mindestens ein vortrainiertes Vision-Sprache-Modell 210 und mindestens ein Maschinenlernmodell 220. Das vortrainierte Vision-Sprache-Modell 210 wird vortrainiert, um Bildklassifikation mit offenen Vokabularklassen durchzuführen. Zum Beispiel umfasst das vortrainierte Vision-Sprache-Modell 210 das Analysenetzwerk Pyramid Scene (PSPNet oder ein semantisches Segmentierungsmodell, das ein Pyramiden-Analysemodul verwendet) mit einem gedehnten Rest-Neuronalnetz (z. B. ResNet50 oder ein Faltungs-Neuronalnetz mit 50 Schichten). Zum Beispiel umfasst das vortrainierte Vision-Sprache-Modell 210 CLIP. In **Fig. 2A** umfasst zum Beispiel das vortrainierte Vision-Sprache-Modell 210 einen modifizierten Bildcodierer 210A (z. B. MaskCLIP), der dafür ausgelegt ist, direkt Pixeleinbettungen, die mit Texten aus dem vortrainierten Vision-Sprache-Modell 210 (z. B. CLIP) korreliert sind, zu extrahieren. Außerdem umfasst, wie in **Fig. 3** und **Fig. 5** gezeigt, das vortrainierte Vision-Sprache-Modell 210 einen Textcodierer 210B, der dafür ausgelegt ist, als Reaktion auf Empfang von einem oder mehreren Text-/Zeichendaten (z. B. Klassennamen) als Eingabe Texteinbettungen als Ausgabe zu erzeugen.

[0013] Zusätzlich ist das Maschinenlernmodell 220 ausgelegt zum Erzeugen von Pixeleinbettungen als Ausgabe bei Empfang mindestens eines Digitalbildes als Eingabe. Das Maschinenlernmodell 220 umfasst mindestens ein CNN (Faltungs-Neuronalnetz), ein beliebiges Codierungsnetzwerk, ein beliebiges geeignetes künstliches Neuronalnetzmodell oder eine beliebige Anzahl und Kombination davon. Zum Beispiel umfasst in **Fig. 2A** das Maschinenlernmodell 220 mindestens einen Codierer 220A. Außerdem umfassen die Maschinenlernenden 130 Sensordaten, Quellenbilddaten, verschiedene Ergänzungsbilddaten, verschiedene Ergänzungsdaten, verschiedene Klassenprototyp-Daten, verschiedene Verlustdaten, verschiedene Gewichtsdaten, verschiedene Parameterdaten und beliebige relevante Daten, die es dem Rahmen 200 ermöglichen, sprachengeführte selbstbeaufsichtigte semantische Segmentierung wie in der vorliegenden Offenbarung beschrieben zu lernen und/oder auszuführen. In der Zwischenzeit stellen die anderen relevanten Daten 140 verschiedene Daten (z. B. Betriebssystem

usw.) bereit, die es dem Rahmen 200 ermöglichen, die Funktionen wie hier besprochen auszuführen.

[0014] Das System 100 kann mindestens ein Sensorsystem 150 umfassen. Das Sensorsystem 150 umfasst einen oder mehrere Sensoren. Zum Beispiel umfasst das Sensorsystem 150 einen Bildsensor, eine Kamera, einen Radarsensor, einen LIDAR-Sensor (Light Detection And Ranging), einen Wärmesensor, einen Ultraschallsensor, einen Infrarotsensor, einen Bewegungssensor, einen Audiosensor, eine IMU (Trägheitsmesseinheit), einen beliebigen geeigneten Sensor oder eine beliebige Anzahl und Kombination davon. Das Sensorsystem 150 ist betreibbar zur Kommunikation mit einer oder mehreren anderen Komponenten (z. B. dem Verarbeitungssystem 110 und dem Speichersystem 120) des Systems 100. Zum Beispiel kann das Sensorsystem 150 Sensordaten bereitstellen, die das Verarbeitungssystem 110 dann zum Erzeugen von Bilddaten (z. B. Digitalbildern) auf der Basis der Sensordaten verwendet. In dieser Hinsicht ist das Verarbeitungssystem 110 ausgelegt zum Erhalten der Sensordaten direkt oder indirekt von einem oder mehreren Sensoren des Sensorsystems 150. Das Sensorsystem 150 ist lokal, entfernt oder eine Kombination davon (z. B. teilweise lokal und teilweise entfernt). Auf den Empfang der Sensordaten hin ist das Verarbeitungssystem 110 ausgelegt zum Verarbeiten dieser Sensordaten und/oder Erzeugen von Bilddaten (z. B. Digitalbildern) in Verbindung mit dem Rahmen 200, den Maschinenlernenden 130, den anderen relevanten Daten 140, beliebigen geeigneten Software- und/oder Hardwarekomponenten oder einer beliebigen Anzahl und Kombination davon.

[0015] Zusätzlich kann das System 100 mindestens eine andere Komponente umfassen. Zum Beispiel ist, wie in **Fig. 1** gezeigt, das Speichersystem 120 auch ausgelegt zum Speichern anderer relevanter Daten 140, die den Betrieb des Rahmens 200 in Bezug auf eine oder mehrere Komponenten (z. B. das Sensorsystem 150, E/A-Vorrichtungen 160 und andere Funktionsmodule 170) betreffen. Zusätzlich ist das System 100 dafür ausgelegt, eine oder mehrere E/A-Vorrichtungen 160 (z. B. eine Anzeigevorrichtung, eine Tastaturvorrichtung, eine Lautsprechervorrichtung usw.) zu umfassen, die den Rahmen 200 betreffen. Außerdem umfasst das System 100 andere Funktionsmodule 170, wie etwa beliebige entsprechende Hardware, Software oder eine Kombination davon, die bei dem Funktionieren des Rahmens 200 helfen oder dazu beitragen. Zum Beispiel umfassen die anderen Funktionsmodule 170 Kommunikationstechnologie, die es Komponenten des Systems 100 ermöglicht, wie hier beschrieben miteinander zu kommunizieren. In dieser Hinsicht ist das System 100 betreibbar, um mindestens den Rahmen 200 für sprachengeführte selbstbeaufsichtigte semantische Segmentierung wie hier

beschrieben zu trainieren, zu verwenden und/oder einzusetzen.

[0016] Fig. 2A, Fig. 2B, Fig. 2C und Fig. 2D zeigen verschiedene Aspekte eines Beispiels für den Rahmen 200 für sprachengeführte selbstbeaufsichtigte semantische Segmentierung (im folgenden als CLIP-S4 bezeichnet), wobei sich „CLIP“ auf kontrastives Sprache-Bild-Vortraining („CLIP“) bezieht und sich „S4“ auf die „selbstbeaufsichtigte semantische Segmentierung“ bezieht. Wie erwähnt, umfasst der Rahmen 200 ein vortrainiertes Vision-Sprache-Modell 210 und ein Maschinenlernmodell 220. Ein oder mehrere Prozessoren des Verarbeitungssystems 110 sind dafür ausgelegt, den Rahmen 200 auszuführen.

[0017] Bei einer beispielhaften Ausführungsform umfasst das vortrainierte Vision-Sprache-Modell 210 mindestens ein CLIP-Modell. In dieser Hinsicht verwendet der Rahmen 200 direkt das vortrainierte Vision-Sprache-Modell, das für Klassifikationsaufgaben vortrainiert wird. Zum Beispiel umfasst in Fig. 2A der Rahmen 200 mindestens einen modifizierten Bildcodierer 210A (z. B. MaskCLIP) des vortrainierten Vision-Sprache-Modells 210 (z. B. CLIP). Der modifizierte Bildcodierer 210A ist dafür ausgelegt, direkt Pixeleinbettungen, die mit Texten aus dem vortrainierten Vision-Sprache-Modell 210 (z. B. dem CLIP-Modell) korreliert sind, zu extrahieren. Zusätzlich umfasst der Rahmen 200 mindestens einen Textcodierer 210B (Fig. 3 und Fig. 5). Das vortrainierte Vision-Sprache-Modell 210 ist ausgelegt zum Empfangen einer Trainingsdatenmenge von Trainingsbildern 10, die Quellenbilder 12 umfassen. Die Quellenbilder 12 beziehen sich auf die „Originalbilder“ der Trainingsdatenmenge und/oder der Trainingsbilder 10 ohne jegliche Datenergänzung (z. B. Umklappen, Abschneiden usw.). In diesem Beispiel sind die Quellenbilder 12 Digitalbilder. Als ein nicht einschränkendes Beispiel ist in Fig. 2A das Quellenbild 12 ein Digitalbild, das einen Jet und einen Mond mit Bezug auf einen Hintergrund eines Himmels zeigt. In dieser Hinsicht ist, wie in Fig. 2A gezeigt, der modifizierte Bildcodierer 210A ausgelegt zum Erzeugen von Pixeleinbettungen als Ausgabe als Reaktion auf Empfang eines oder mehrerer Digitalbilder (z. B. Quellenbild 12) als Eingabe.

[0018] Außerdem ist das Maschinenlernmodell 220 ausgelegt zum Erzeugen von Pixeleinbettungen als Ausgabe bei Empfang eines oder mehrerer Digitalbilder (z. B. Bild 14 und Bild 16) als Eingabe. In dieser Hinsicht ist das Maschinenlernmodell 220 ausgelegt zum Abbilden von Bilddaten auf Pixeleinbettungen zur semantischen Segmentierung. Zum Beispiel umfasst in Fig. 2A das Maschinenlernmodell 220 ein Siam-Codierungsnetz. Das Maschinenlernmodell 220 umfasst mindestens einen Codierer 220A. Der Codierer 220A kann ein CNN (Faltungs-Neuronal-

netz), ein beliebiges geeignetes Codierungsnetz, ein beliebiges geeignetes Maschinenlernmodell oder eine beliebige Anzahl und Kombination davon umfassen.

[0019] Fig. 2A zeigt ein Beispiel für den Rahmen 200, der mindestens die folgenden zwei Komponenten umfasst: (i) eine Komponente 230 für selbstbeaufsichtigtes kontrastives Lernen und (ii) eine Komponente 240 für durch ein Vision-Sprache-Modell geführte Einheitlichkeit. Speziell erzwingt die Komponente 230 für selbstbeaufsichtigtes kontrastives Lernen über eine erste Verlustkomponente 270 des Maschinenlernmodells 220, dass Pixeleinbettungen in visuell kohärenten Regionen und unter verschiedenen ergänzten Ansichten (z. B. Bild 14 und Bild 16) desselben Quellenbildes 12 einheitlich sind. Zum Beispiel umfasst in Fig. 2A die Komponente 230 für selbstbeaufsichtigtes kontrastives Lernen mindestens eine erste Verlustkomponente 270, um diese kontrastive Einheitlichkeit zu erzielen.

[0020] Die Komponente 240 für durch ein Vision-Sprache-Modell geführte Einheitlichkeit führt Folgendes ein: (i) Einbettungs-Einheitlichkeit über eine zweite Verlustkomponente 280 des Maschinenlernmodells 220, sodass die durch das Maschinenlernmodell 220 erzeugten Pixeleinbettungen mit dem Raum gemeinsamer Merkmale von Texten und Bildern, die durch das vortrainierte Vision-Sprache-Modell 210 (z. B. CLIP) bereitgestellt werden, ausgerichtet sind, und (ii) semantische Einheitlichkeit über eine dritte Verlustkomponente 290 des Maschinenlernmodells 220, sodass das Maschinenlernmodell 220 dazu gezwungen wird, dieselben Vorhersagen wie das vortrainierte Vision-Sprache-Modell 210 für eine Menge von Zielklassen mit sowohl bekannten Klassenprototypen 262 als auch unbekannten Klassenprototypen 264 zu tätigen. Für bekannte Klassen sind die bekannten Klassenprototypen 262 im Voraus berechnet und während des Trainings fest. Für unbekannte Klassen werden die unbekannten Klassenprototypen 264 während des Trainings über Clusterung gelernt. In dieser Hinsicht sind die Komponente 230 für selbstbeaufsichtigtes kontrastives Lernen und die Komponente 240 für durch ein Vision-Sprache-Modell geführte Einheitlichkeit zueinander komplementär. Diese zwei Komponenten verbessern die Pixeleinbettungsqualität und ermöglichen sprachengeführte semantische Segmentierung sowohl für bekannte Klassen (z. B. eine Menge von Klassen, die vordefiniert und durch den Rahmen 200 als eine Menge von Texteinbettungen während des Trainings verwendet werden) als auch für unbekannte Klassen (z. B. eine andere Menge von Klassen, die nicht vordefiniert sind und somit durch den Rahmen 200 während des Trainings gelernt werden).

[0021] In **Fig. 2A** segmentiert der Rahmen 200 Digitalbilder, indem bewirkt wird, dass das Maschinenlernmodell 220 eine Pixeleinbettungsfunktion mit (i) selbstbeaufsichtigtem kontrastivem Lernen und (ii) der Führung eines vortrainierten Vision-Sprache-Modells 210 (z. B. CLIP) lernt. Der Rahmen 200 verwendet selbstbeaufsichtigtes kontrastives Lernen, um die Einheitlichkeit von Pixeleinbettungen in visuell kohärenten Regionen (z. B. Superpixeln) und unter verschiedenen ergänzten Ansichten (z. B. Bild 14 und Bild 16), die von dem selben Quellenbild 12 erhalten werden, zu erzwingen. Der Rahmen 200 führt zwei Komponenten für durch Vision-Sprache-Modell geführte Einheitlichkeit ein (d.h., Einbettungs-Einheitlichkeit und semantische Einheitlichkeit), um das Maschinenlernmodell 220 weiter zu regularisieren. Diese zwei Komponenten sind zueinander komplementär. Außerdem mindert kontrastives Lernen das durch das vortrainierte Vision-Sprache-Modell 210 (z. B. CLIP) eingeführte Rauschen. Zusätzlich ist mit dem aus dem vortrainierten Vision-Sprache-Modells 210 (z. B. CLIP) extrahierten Wissen der Rahmen 200 ausgelegt zum Verbessern der Qualität der Pixeleinbettungen. Der Rahmen 200 ist ausgelegt zum Durchführen von sprachengeführter semantischer Segmentierung mit Ziel-Klassenprototypen 260, die sowohl bekannte Klassenprototypen 262 als auch unbekannte Klassenprototypen 264 umfassen.

[0022] Der Rahmen 200 trainiert eine Pixeleinbettungsfunktion, einheitliche Pixeleinbettungen in visuell kohärenten Regionen zu erzeugen, mittels kontrastivem Pixelsegment-Lernen. Speziell verwendet der Rahmen 200 eine Einbettungsfunktion, die jedes Pixel p eines Digitalbildes in einen Einheitslängen-Einbettungsvektor z_p der Dimension d transformiert, über ein tiefes Neuronales Netz, wie etwa das Maschinenlernmodell 220. Zum Beispiel erzeugt in **Fig. 2A** das Maschinenlernmodell 220 eine Menge von Pixeleinbettungen, wie in Bild 22 gezeigt, auf der Basis der Pixel des Bildes 14. Speziell erzeugt der Codierer 220A Pixeleinbettungen als Ausgabe bei Empfang des Bildes 14 als Eingabe. In dieser Hinsicht wird das Bild 22 als eine Visualisierung der Pixeleinbettungen bereitgestellt, die durch das Maschinenlernmodell 220 (z. B. den Codierer 220A) auf der Basis des Bildes 14 erzeugt werden. Das Bild 14 ist eine modifizierte Version des Quellenbildes 12 auf der Basis von Datenergänzung. Als nichteinschränkendes Beispiel wird in **Fig. 2A** das Bild 14 mindestens durch Ausführen von Farb-Raum-Transformationen des Quellenbildes 12 erzeugt, dergestalt, dass das Bild 14 eine andere farbige Version des Quellenbildes 12 ist.

[0023] Zusätzlich erzeugt das Maschinenlernmodell 220 eine Menge von Pixeleinbettungen, wie in Bild 24 gezeigt, auf der Basis des Bildes 16. Speziell erzeugt der Codierer 220A Pixeleinbettungen als

Ausgabe bei Empfang des Bildes 16 als Eingabe. In dieser Hinsicht wird das Bild 24 als eine Visualisierung der Pixeleinbettungen bereitgestellt, die durch das Maschinenlernmodell 220 (z. B. den Codierer 220A) auf der Basis des Bildes 16 erzeugt werden. Das Bild 16 ist eine weitere modifizierte Version des Quellenbildes 12 auf der Basis von Datenergänzung. Als nichteinschränkendes Beispiel wird in **Fig. 2A** das Bild 16 erzeugt mindestens durch Ausführen einer Umklappoperation an dem Quellenbild 12, dergestalt, dass das Bild 16 eine umgeklappte Version des Quellenbildes 12 ist.

[0024] Der Rahmen 200 teilt die Pixeleinbettungen dann durch Clustern der Pixeleinbettungen zu einer Menge von Clustern in Segmente $|S|$ auf. Zum Beispiel erzeugt in **Fig. 2A** mit Bezug auf die Pixeleinbettungen, gezeigt in Bild 22, der Rahmen 200 eine Menge von Clustern, wie definiert durch Segmentkonturdaten und wie durch die in Bild 32 gezeigten Clusterumrisse gezeigt. In **Fig. 2A** wird das Bild 32 als eine Visualisierung der Segmentkonturdaten bereitgestellt, die auf der Basis der Pixeleinbettungen erzeugt werden, die in Bild 22 gezeigt sind. Außerdem erzeugt der Rahmen 200 mit Bezug auf die Pixeleinbettungen des Bildes 24 eine Menge von Clustern, die durch die Segmentkonturdaten definiert und von den Clusterumrissen, die in Bild 34 gezeigt sind, gezeigt sind. In **Fig. 2A** ist das Bild 34 als eine Visualisierung der Segmentkonturdaten bereitgestellt, die basierend auf den in Bild 24 gezeigten Pixeleinbettungen erzeugt werden. Zusätzlich berechnet der Rahmen 200 eine Einbettung v_s jedes Segments (oder eines Einbettungssegments) als den Mittelwert der Pixeleinbettungen dergestalt, dass $v_s = \sum_{p \in s} z_p / |s|$ ist, was auch zu einem Vektor $v_s = v_s / \|v_s\|$ mit Einheitslänge normiert wird.

[0025] Außerdem werden für jedes Pixel p die Segmente in zwei Mengen gruppiert, die eine positive Menge S^+ und eine negative Menge S^- umfassen. Die positive Menge S^+ eines Pixels enthält Segmente mit derselben visuell kohärenten Region des Pixels. Die visuell kohärente Region kann aus Superpixeln oder Konturen abgeleitet werden. Wie erwähnt, verwendet der Rahmen 200 Datenergänzung (z. B. zufällige Umbemessung, Beschneidung, Umklappen, Farb-Jittern, Gaußsche Unschärfe usw.), um einheitliche Pixeleinbettungen zwischen verschiedenen ergänzten Ansichten (z. B. Bild 14 und Bild 16) desselben Quellenbildes 12 zu erzeugen. Daher werden Segmente in derselben visuell kohärenten Region des Pixels in beliebigen der ergänzten Ansichten als die positive Menge S^+ betrachtet. Andere Segmente in dem Bild und andere Segmente aus anderen Bildern im selben Batch sind in der negativen Menge S^- enthalten.

[0026] **Fig. 2B** zeigt ein nicht einschränkendes Beispiel für eine positive Menge 272, die einen Teil einer

Jet-Segmenteinbettung 44A, wie in Bild 44 gezeigt, und einen anderen Teil derselben Jet-Segmenteinbettung 44A, wie in Bild 44 gezeigt, umfasst. Zusätzlich zeigt **Fig. 2B** ein Beispiel für eine weitere positive Menge 274, die die Jet-Segmenteinbettung 44A, wie in Bild 44 gezeigt, und eine weitere Jet-Segmenteinbettung 46A, wie in Bild 46 gezeigt, umfasst. **Fig. 2B** zeigt außerdem ein Beispiel für eine negative Menge 276, die eine unbekannte Segmenteinbettung 44B, wie in Bild 44 gezeigt, und die Jet-Segmenteinbettung 46A, wie in Bild 46 gezeigt, umfasst.

[0027] Die Pixeleinbettung z_p wird dann zu den Segmenten in der positiven Menge S^+ angezogen und von den Segmenten in der negativen Menge S^- über kontrastiven Verlust abgestoßen, wie in Gleichung 1 ausgedrückt. Spezieller ist k in Gleichung 1 die Konzentrationskonstante und $\text{sim}(z_p, v_s)$ ist die Kosinus-ähnlichkeit zwischen der Pixeleinbettung z_p und der Segmenteinbettung v_s . In dieser Hinsicht berechnet der Rahmen 200 Ähnlichkeitsdaten zwischen der Pixeleinbettung z_p und der Segmenteinbettung v_s , wie in Gleichung 1 ausgedrückt, um kontrastive Verlustdaten zu erzeugen.

$$\mathcal{L}_t(p) = -\log \frac{\sum_{s \in S^+} \exp(\text{sim}(z_p, v_s) \kappa)}{\sum_{s \in S^+ \cup S^-} \exp(\text{sim}(z_p, v_s) \kappa)} \quad [1]$$

[0028] Außerdem erhält, wie in **Fig. 2A** gezeigt, der Rahmen 200 die Pixeleinbettungen des Quellenbildes 12 aus dem vortrainierten Vision-Sprache-Modell 210 über den modifizierten Bildcodierer 210A (z. B. MaskCLIP). Der modifizierte Bildcodierer 210A umfasst eine Modifikation der auf Aufmerksamkeit basierenden Pooling-Schicht des Bildcodierers des vortrainierten Vision-Sprache-Modells 210. Spezieller umfasst diese Modifikation (i) Entfernung der Anfrage- und Schlüsselschutzschichten und (ii) eine Umformulierung der Wertprojektionsschicht und der letzten linearen Schicht als zwei aufeinanderfolgende vollständig verbundene Schichten. Beispielsweise verwendet zum Beispiel der Rahmen 200 clip-i (·) als den modifizierten Bildcodierer 210A und clip-t (·) als Textcodierer 210B. Wieder mit Bezug auf **Fig. 2A** sind die Pixeleinbettungen, die durch den modifizierten Bildcodierer 210A erzeugt werden und die in Bild 20 gezeigt sind, gröber und verrauschter als die Pixeleinbettungen, die durch den Codierer 220A erzeugt werden und die in Bild 22 bzw. Bild 24 gezeigt sind.

[0029] Der Rahmen 200 erhält dann die Pixeleinbettungen, die durch das vortrainierte Vision-Sprache-Modell 210 erzeugt wurden, für verschiedene ergänzte Ansichten des Quellenbildes 12. Speziell erzeugt der Rahmen 200 über das vortrainierte Vision-Sprache-Modell 210 die Pixeleinbettungen, wie in Bild 20 gezeigt, als Ausgabe bei Empfang des Quellenbildes 12 als Eingabe. In **Fig. 2A** wird

das Bild 20 als eine Visualisierung der Pixeleinbettungen bereitgestellt, die durch das vortrainierte Vision-Sprache-Modell 210 (z. B. den modifizierten Bildcodierer 210A, wie etwa MaskCLIP) erzeugt werden.

[0030] Der Rahmen 200 ist dafür ausgelegt, danach Datenergänzung durchzuführen, um sicherzustellen, dass die Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 unter verschiedenen ergänzten Ansichten den Pixeleinbettungen des Maschinenlernmodells 220 entsprechen. Der Rahmen 200 minimiert außerdem die Distanz von Einbettungen zwischen Segmenten (anstatt Pixeln) aus den selbstbeaufsichtigten und Vision-Sprache-Einbettungsräumen. Wie in **Fig. 2A** gezeigt, werden diese Aktionen an Segmenteinbettungen ausgeführt, weil die Segmenteinbettungen rauschbehaftet sind, die durch Aggregieren über Segmente gemindert werden. Daher verwendet der Rahmen 200 die durch das Maschinenlernmodell 220 erzeugten Pixeleinbettungen zum Ableiten von Segmenten (oder Clustern) visuell kohärenter Regionen und Erzeugen von Segmentkonturdaten, die diese Segmente (oder Cluster) definieren.

[0031] Der Rahmen 200 umfasst einen Segmentapplikator 250. Der Segmentapplikator 250 ist ausgelegt zum (i) Empfangen der Datenergänzungsdaten, die das Bild 14 aus dem Quellenbild 12 erzeugten, und der Segmentkonturdaten, die die Cluster des Bildes 32 definieren, und (ii) Erzeugen von Segmenteinbettungen des Bildes 40 durch Anwenden der Datenergänzungsdaten und der Segmentkonturdaten auf die Pixeleinbettungen des Bildes 20. Zusätzlich ist der Segmentapplikator 250 ausgelegt zum (i) Empfangen der Datenergänzungsdaten, die das Bild 16 aus dem Quellenbild 12 erzeugten, und der Segmentkonturdaten, die die Cluster des Bildes 34 definieren, und (ii) Erzeugen von Segmenteinbettungen des Bildes 42 durch Anwenden der Datenergänzungsdaten und der Segmentkonturdaten auf die Pixeleinbettungen des Bildes 20.

[0032] Für jedes Segment s definiert der Rahmen 200 den einbettungseinheitlichen Verlust über Gleichung 2, wobei v_s und i_s die Segmenteinbettungen sind, die aus der Einbettungsfunktion des Maschinenlernmodells 220 bzw. der Einbettungsfunktion des vortrainierten Vision-Sprache-Modells 210 (z. B. CLIP) abgeleitet werden. Hier ist der Mittelwert der Pixeleinbettungen des vortrainierten Vision-Sprache-Modells aus Segment s nämlich $i_s = \sum_{p \in s} \text{clip-i}(s)/|s|$. Der Rahmen 200 erzeugt einbettungseinheitliche Verlustdaten über Gleichung 2, was Berechnen von Ähnlichkeitsdaten über eine Kosinusähnlichkeit zwischen der Segmenteinbettung v_s des Maschinenlernmodells 220 und der Segmenteinbettung i_s des vortrainierten Vision-Sprache-Modells 210 umfasst.

$$\mathcal{L}_e(s) = 1 - \text{sim}(v_s, i_s) \quad [2]$$

[0033] Fig. 2C zeigt Aspekte der Berechnung des einbettungseinheitlichen Verlusts. In diesem nicht einschränkenden Beispiel erzeugt der Rahmen 200 einbettungseinheitliche Verlustdaten durch Vergleichen einer Ähnlichkeit zwischen einer Segmenteinbettung 46A, die aus dem Maschinenlernmodell 220 abgeleitet wird, und einer entsprechenden Segmenteinbettung 42A, die aus dem vortrainierten Vision-Sprache-Modell 210 abgeleitet wird. Wie in Fig. 2C gezeigt, hat die Segmenteinbettung 46A aus dem Maschinenlernmodell 220 höhere Qualität als die Segmenteinbettung 42A des vortrainierten Vision-Sprache-Modells 210. Die Segmenteinbettung 46A, die in Bild 46 gezeigt ist, und die Segmenteinbettung 42A, die in Bild 42 gezeigt ist, entsprechen beide einer visuell kohärenten Region oder einem Segment in Bezug auf eine bekannte Klasse von Jet.

[0034] Zusätzlich zur Einbettungseinheitlichkeit stellt der Rahmen 200 semantische Einheitlichkeit bereit, indem das Maschinenlernmodell 220 gezwungen wird, dieselben Vorhersagen semantischer Klassen wie das vortrainierte Vision-Sprache-Modell 210 zu tätigen. Die Argumentation ist, dass der Rahmen 200 dafür ausgelegt ist, bessere Pixeleinbettungen zu erzeugen, wenn unterscheidbare Cluster gebildet werden, so dass sie verschiedenen semantischen Klassen entsprechen, da das Ziel semantischer Segmentierung ist, pixelweise Klassifikation durchzuführen. Semantische Einheitlichkeit wird über Pseudokennzeichnung erzielt. Wieder erzwingt der Rahmen 200 die semantische Einheitlichkeit direkt auf der Segmentebene (nicht direkt auf der Pixel-ebene), um das Rauschen in Pseudokennzeichnungen y_s zu verringern.

[0035] Speziell verwendet für jedes Segment s der Rahmen 200 das vortrainierte Vision-Sprache-Modell 210 zur Erzeugung seiner Pseudokennzeichnung y_s über eine Menge von Zielklassen, die sowohl bekannte Klassen als auch unbekannte Klassen umfassen. Die Zielklassen werden auch später in der vorliegenden Offenbarung besprochen. Die Pseudokennzeichnung y_s wird auf der Basis der größten Ähnlichkeit oder der höchsten Ähnlichkeit zwischen der Segmenteinbettung i_s mit einer Menge von Klassenprototypen 260 $C = \{c_l\}_0^{L-1}$ der Zielklassen im Pixeleinbettungsraum des vortrainierten Vision-Sprache-Modells 210 dergestalt erzeugt, dass $y_s = \arg\max_{l \in L} (\text{sim}(i_s, c_l))$ ist.

[0036] Dann definiert, wie in Gleichung 3 ausgedrückt, der Rahmen 200 den semantikeinheitlichen Verlust als die Kreuzentropie zwischen der Vorhersage $\phi(v_s)$ durch das Maschinenlernmodell 220 über die Zielklassen und die Pseudokennzeichnung y_s hinweg. In Gleichung 3 ist $\phi(v_B) = \text{softmax}(\text{sim}(v_s, C))$.

$(v_s, C))$. Der Rahmen 200 erzeugt semantikeinheitliche Verlustdaten über Gleichung 3.

$$\mathcal{L}_e(s) = H(y_s, \phi(v_s)) \quad [3]$$

[0037] Fig. 2D zeigt Aspekte der Berechnung des semantikeinheitlichen Verlusts unter Verwendung von Ziel-Klassenprototypen 260. Spezieller vergleicht in dem in Fig. 2D gezeigten nicht einschränkenden Beispiel der Rahmen 200 eine Ähnlichkeit zwischen einer Segmenteinbettung 42A, die aus dem vortrainierten Vision-Sprache-Modell 210 abgeleitet wird, und jedem der Ziel-Klassenprototypen 260, wozu bekannte Klassenprototypen 262 und unbekannte Klassenprototypen 264 gehören. Der Rahmen 200 erzeugt Ähnlichkeitsdaten 296 über eine Kosinusähnlichkeitsfunktion (bezeichnet als „Sim (erstes Merkmal, zweites Merkmal)“). Die Ähnlichkeitsdaten 296 geben ein Maß dafür an, wie ähnlich (oder unähnlich) die Segmenteinbettung 42A jedem der Ziel-Klassenprototypen 260 ist. Wie bereits erwähnt, besteht größere Ähnlichkeit zwischen zwei Merkmalen, wenn der Ähnlichkeitswert 1 näher oder gleich ist. Außerdem besteht größere Unähnlichkeit zwischen zwei Merkmalen, wenn die Ähnlichkeit 0 näher oder gleich ist. Zum Beispiel erzeugt in diesem nicht einschränkenden Beispiel der Rahmen 200 Ähnlichkeitsdaten 296, die angeben, dass die Segmenteinbettung 42A dem bekannten Klassenprototypen von Jet verglichen mit den anderen bekannten Klassenprototypen und unbekannten Klassenprototypen ähnlicher ist. Der Rahmen 200 erzeugt außerdem zusätzliche Daten 294, um die Klasse mit der größten vorhergesagten Wahrscheinlichkeit über eine argmax-Funktion zu finden. In diesem Fall bestimmt der Rahmen 200, dass der Klassenprototyp von Jet verglichen mit den anderen Klassenprototypen (z. B. Klassenprototyp von Katze, ein Klassenprototyp einer ersten unbekannten Klasse, ein Klassenprototyp einer zweiten unbekannten Klasse usw.) die größte vorhergesagte Wahrscheinlichkeit für die Segmenteinbettung 42A aufweist. Der Rahmen 200 verwendet deshalb den bekannten Klassenprototypen von Jet als die Pseudokennzeichnung y_s über die Menge von Ziel-Klassenprototypen 260.

[0038] Zusätzlich vergleicht der Rahmen 200 eine Ähnlichkeit zwischen einer Segmenteinbettung 46A, die aus dem Maschinenlernmodell 220 abgeleitet wird, und jedem der Ziel-Klassenprototypen 260, wozu bekannte Klassenprototypen 262 und unbekannte Klassenprototypen 264 gehören. Der Rahmen 200 erzeugt Ähnlichkeitsdaten 292 über eine Kosinusähnlichkeitsfunktion (bezeichnet als „Sim (erstes Merkmal, zweites Merkmal)“). Die Ähnlichkeitsdaten 292 (bzw. Ähnlichkeitsmessung) geben ein Maß dafür an, wie ähnlich (oder unähnlich) die Segmenteinbettung 46A jedem der Ziel-Klassenprototypen 260 ist. Es besteht größere Ähnlichkeit zwischen zwei Merkmalen, wenn der Ähnlichkeitswert 1

näher oder gleich ist. Außerdem besteht größere Unähnlichkeit zwischen zwei Merkmalen, wenn die Ähnlichkeit 0 näher oder gleich ist. Zum Beispiel erzeugt in diesem nicht einschränkenden Beispiel der Rahmen 200 Ähnlichkeitsdaten, die angeben, dass die Segmenteinbettung 46A dem bekannten Klassenprototypen von Jet verglichen mit den anderen bekannten Klassenprototypen und unbekannten Klassenprototypen ähnlicher ist. Der Rahmen 200 stellt den bekannten Klassenprototyp von Jet als die Klassenvorhersage über die Zielklassen bereit. Der Rahmen verwendet dann Gleichung 3 und berechnet den semantikeinheitlichen Verlust als die Kreuzentropie zwischen der Vorhersage-Klassenvorhersage des Maschinenlernmodells 220 und der Pseudokennzeichnung y_s .

[0039] Fig. 3 ist ein Flussdiagramm 300, das Aspekte von Ziel-Klassenprototypen 260 darstellt. Wie in Fig. 3 gezeigt, umfassen die Ziel-Klassenprototypen 260 sowohl bekannte Klassenprototypen 262 als auch unbekannte Klassenprototypen 264, repräsentiert durch $C = \{C_k, C_u\}$, wobei C eine Menge der Ziele-Klassenprototypen 260 repräsentiert, C_k eine Menge der bekannten Klassenprototypen 262 repräsentiert und C_u eine Menge der unbekannten Klassenprototypen 264 repräsentiert. Für eine Menge bekannter Klassen 302 (z. B. Jet, Katze usw.) mit $K = \{0, \dots, k - 1\}$ erzeugt der Rahmen 200 zuerst über einen Textcodierer 210B des vortrainierten Vision-Sprache-Modells 210 eine Menge von Texteinbettungen 306 $T = \{t_k\} = \{\text{clip} - t(k)\}$ durch Verwendung einer Menge von Aufforderungsvorlagen 304. Insbesondere empfängt der Textcodierer 210B die Menge bekannter Klassen als Texteingang über die Aufforderungsvorlagen 304. Dann berechnet der Rahmen 200 über Softmax die normierte Ähnlichkeit zwischen den Texteinbettungen 306 T und allen Segmenteinbettungen 308 $I = \{i_s\}$ des vortrainierten Vision-Sprache-Modells 210 auf der Basis der Trainingsbilder 10 und Mittelwert der obersten m ähnlichen Segmenteinbettungen 308 des vortrainierten Vision-Sprache-Modells 210 als Klassenprototyp für

$$C_k = [c_k] = \text{avg}_m \left(\text{top} - m_s \left(\text{softmax}_k \left(\text{sim}(I, T) \right) \right) \right).$$

diese Klasse.

[0040] Für jede unbekannte Klasse c_u wählt der Rahmen 200 zufällig eine Segmenteinbettung 308 (z. B. i_i) eines Bildsegments als den anfänglichen Prototyp $c_u = i_r$. Die zufällige Auswahl kann mit Bezug auf eine der Menge aller Segmenteinbettungen 308 entnommene Teilmenge erfolgen. Die Teilmenge umfasst Segmenteinbettungen, die nicht der Menge bekannter Klassen 302 zugeordnet sind. Zum Beispiel ist i_0 der bekannten Klasse von Jet zugeordnet, während i_3 nicht einer bekannten Klasse aus der Menge bekannter Klassen 302 zugeordnet ist. Anders ausgedrückt, umfasst die Teilmenge eine

Anzahl von Segmenteinbettungen 308, die unbekannten Klassen (z. B. aktuell unbenannten Klassen, die Platzhalterklassen zugeordnet sind) entsprechen. Zum Beispiel wählt in Fig. 3 der Rahmen 200 zufällig ein Einbettungssegment i_3 aus dieser Teilmenge dergestalt aus, dass $i_r = i_3$ ist. Statt Texteinbettungen 306 von nur einer Menge vordefinierter/-bekannter Klassen 302 zu verwenden, umfasst der Rahmen 200 zwei Mengen von Zielklassen sowohl mit bekannten Klassenprototypen 262 als auch unbekannten Klassenprototypen 264, wodurch sprachengeführte klassenlose semantische Segmentierung ermöglicht wird.

[0041] Der Entwurf von Zielklassen und zugeordneten Klassenprototypen $C = \{c_l\}_0^{L-1}$ ist zur Erzielung der semantischen Einheitlichkeit entscheidend. Hier ist ein Klassenprototyp ein Einbettungsvektor, der eine Klasse in einem Einbettungsraum (z. B. Vision-Sprache-Einbettungsraum) repräsentieren kann. Zum Beispiel kann der Klassenprototyp der mittlere Vektor von Einbettungen aller Segmente einer Klasse „Auto“ (oder der mittlere Vektor aller Segmenteinbettungen einer Klasse „Auto“) sein. Anzunehmen, dass die Zielklassen bereits vordefiniert sind, ist in realen Benutzungsfällen ohne menschliche Annotationen nicht durchführbar. Solche Verfahren mit diesen Annahmen können somit mit unbekannten Klassen, die in den Daten verborgen sind, nicht umgehen. Um dieses Problem anzugehen, umfasst der Rahmen 200 zwei Mengen von Klassenprototypen 260. Die erste Menge von Klassenprototypen 260 umfasst bekannte Klassenprototypen 262 $C_k = \{c_0, \dots, c_{k-1}\}$, die der Menge bekannter Klassen 302 zugeordnet sind. Die Menge bekannter Klassen 302 ist vorbestimmt und wird dem Rahmen 200 während des Trainings gegeben. Die zweite Menge von Klassenprototypen umfasst unbekannte Klassenprototypen 264 $C_u = \{c_k, \dots, c_{k+u}\}$, die unbekannten Klassen zugeordnet sind. Die unbekannten Klassenprototypen 264 sind unbekannten Klassen (die manchmal als „Platzhalterklassen“ bezeichnet werden) zugeordnet. Die unbekannten Klassen dienen als Platzhalter, bis diese Eingangsdaten zu einem späteren Zeitpunkt (z. B. während Schlussfolgerung) erhalten werden, da diese Eingangsdaten nicht während des Trainings als die Menge bekannter Klassen 302 bereitgestellt werden. Die bekannten Klassenprototypen 262 werden durch wirksame Nutzung des vortrainierten Vision-Sprache-Modells 210 vordefiniert. Unterdessen werden die unbekannten Klassenprototypen 264 während des Trainings aus den Segmenteinbettungen erlernt. Somit hat der Rahmen 200 Ziel-Klassenprototypen 260, die als

$$C = \{c_l\}_0^{L-1} = \{c_0, \dots, c_{k-1}, c_k, \dots, c_{k+u}\} \quad \text{repräsentiert werden können, wobei } L = k + u \text{ ist.}$$

[0042] Für bekannte Klassen ist eine natürliche Wahl, die durch das vortrainierte Vision-Sprache-

Modell 210 erzeugten Texteinbettungen (z. B. Kontur 406) als ihre Klassenprototyp-Einbettungen zu verwenden. Obwohl die Texteinbettungen (z. B. Kontur 406) dafür trainiert sind, sich mit Bild-/Pxeleinbettungen (z. B. Kontur 402) auszurichten, besteht jedoch immer noch eine große Lücke zwischen den Texteinbettungen (z. B. Kontur 406) und den Bild-/Pxeleinbettungen (z. B. Kontur 402) im gemeinsamen Vision-Sprache-Einbettungsraum 400, wie in **Fig. 4A** und **Fig. 4B** gezeigt ist. Deshalb gibt es einige Probleme beim Erlernen bedeutsamer unbekannter Klassen aus Bildmerkmalen bei der Verwendung von Texteinbettungen (z. B. Kontur 402) als Klassenprototypen. Daher nutzt der Rahmen 200 wirksam Pxeleinbettungen aus dem vortrainierten Vision-Sprache-Modell 210, um jede bekannte Klasse zu repräsentieren.

[0043] Hierzu erhält der Rahmen 200 für eine Menge bekannter Klassen 302 (z. B. Jet, Katze usw.), $k = \{0, \dots, k-1\}$, zuerst eine Menge von Texteinbettungen 306 $T = \{t_k\} = \{\text{clip} - t(k)\}$ über eine Menge von Aufforderungsvorlagen 304. Zum Beispiel kann die Menge bekannter Klassen 302 durch einen Benutzer (z. B. Software-Entwickler usw.) über die Aufforderungsvorlagen 304 während des Trainings in das vortrainierte Vision-Sprache-Modell 210 (z. B. CLIP) eingegeben werden. Der Rahmen 200 erhält außerdem eine Menge von Segmenteinbettungen $I = \{i_s\}$, die durch das vortrainierte Vision-Sprache-Modell 210 für alle Trainingsbilder 10 erzeugt werden durch a) Einspeisen von Trainingsbildern 10 in den modifizierten Bildcodierer 210A zur Erzeugung von Pxeleinbettungen; b) Clustern der Pxeleinbettungen als Segmente und c) Mitteln der Pxeleinbettungen in jedem Segment s . Dementsprechend ist der Rahmen 200 ausgelegt zum Erzeugen einer Segmenteinbettung für jedes Segment, wie repräsentiert durch

$$i_s = \sum_{p \in s} \text{clip} - i(s) / |\hat{s}|.$$

[0044] Der Rahmen 200 (i) berechnet die Ähnlichkeit zwischen Texteinbettungen 306 bekannter Klassen, die durch das vortrainierte Vision-Sprache-Modell 210 erzeugt werden, und allen Segmenteinbettungen 308, die durch das vortrainierte Vision-Sprache-Modell 210 erzeugt werden, und (ii) normiert diese Ähnlichkeiten über alle bekannten Klassen durch Softmax. Als letztes mittelt der Rahmen 200 die obersten am ähnlichen Segmenteinbettungen, um einen bekannten Klassenprototypen 262 für jede bekannte Klasse

$C_k = [c_k] = \text{avg}_m(\text{top} - m_s(\text{softmax}_k(\text{sim}(I, T))))$ zu erzeugen. Außerdem werden in **Fig. 3** die unbekannten Klassenprototypen 264 (z. B. Einbettungsvektoren) der unbekannten Klassen C_u zufällig initialisiert durch Stichprobennahme der Segmenteinbettungen 308, die durch das vortrainierte Vision-Sprache-Modell 210 erzeugt werden. Zum Beispiel entnimmt

der Rahmen 200 zufällige Stichproben, sodass $C_u = \text{random}(\text{clip} - i(\hat{S}), u)$ ist, mit einer Größe der unbekannten Klasse von u .

[0045] Während des Trainings wird die Segmenteinbettung c_u jedes unbekannten Klassenprototypen 264 aktualisiert durch Minimieren ihrer Distanz zu allen Segmenten, die als diese unbekannte Klasse klassifiziert werden (ähnlich der Aktualisierung der Schwerpunkte bei der k-means-Clusterung), wie in Gleichung 4 ausgedrückt. In diesem Beispiel verwendet der Rahmen 200 die Kosinusähnlichkeitsfunktion zur Erzeugung von Ähnlichkeitsdaten (oder Distanzdaten) zwischen der Segmenteinbettung c_u und allen Segmenteinbettungen (oder allen Segmenten von Pxeleinbettungen), die als diese unbekannte Klasse klassifiziert werden. Spezieller repräsentiert in Gleichung 4 S_u die Segmente (oder Segmenteinbettungen), die als die unbekannten Klassen klassifiziert werden. In dieser Hinsicht ist der Rahmen 200 ausgelegt zum Erzeugen von Prototyp-Verlustdaten über Gleichung 4 und Lernen der Pixelrepräsentationen unbekannter Klassen. Während des Trainings aktualisiert der Rahmen 200 die Einbettungen für die unbekannten Klassen (oder aktualisiert einen unbekannten Klassenprototypen 204 60) über $\mathcal{L}_u$.

$$\mathcal{L}_u = \sum_{s \in S_u} \left((1 - \text{sim}(c_u, \text{clip} - i(s))) / |S_u| \right) \quad [4]$$

[0046] Kurz gefasst trainiert der Rahmen 200 die Pxeleinbettungsfunktion durch einen kombinierten Verlust $\mathcal{L}$, der eine Summe ist, die die kontrastiven Pixelsegment-Verlustdaten (z. B. $\mathcal{L}_t$), die einbettungseinheitlichen Verlustdaten (z. B. $\mathcal{L}_g$) und semantischen einheitlichen Verlustdaten (z. B. $\mathcal{L}_s$) kombiniert, wie in Gleichung 5 angegeben. In dieser Hinsicht erzeugt der Rahmen 200 kombinierte Verlustdaten über Berechnung des in Gleichung 5 ausgedrückten kombinierten Verlusts $\mathcal{L}$. Zusätzlich ist der Rahmen 200 ausgelegt zum Aktualisieren von Parametern des Maschinenlernmodells 220 auf der Basis dieser kombinierten Verlustdaten.

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_g + \mathcal{L}_s \quad [5]$$

[0047] Zur Schlussfolgerung verwendet das System 100 den trainierten Rahmen 200 (z. B. das trainierte Maschinenlernmodell 220) zum Erzeugen von Pxeleinbettungen für jedes Eingangsbild (z. B. Digitalbild) und verwendet die Pxeleinbettungen für verschiedene Downstream-Aufgaben, darunter sprachengeführte und unbeaufsichtigte semantische Segmentierung. Zur sprachengeführten semantischen Segmentierung erhält der Rahmen 200 zuerst die Texteinbettungen beliebiger Schlussfolgerungsklassen durch Einspeisen der aufforderungstechnischen Texte in Aufforderungsvorlagen 304 für den Textcodierer 210B des vortrainierten Vision-Sprache-Modells 210. Dann versieht der Rahmen 200

jedes Pixel oder Segment mit dem Klassennamen (oder der Klassenkennzeichnung), dessen Texteinbettung der Pixeleinbettung oder Segmenteinbettung, die durch den Rahmen 200 (z. B. das trainierte Maschinenlernmodell 220) erzeugt wird, am nächsten kommt. Zur unbeaufsichtigten semantischen Segmentierung verwendet der Rahmen 200 k-Nächste-Nachbarn-Suche oder einen Linear-Klassifizierer zur Durchführung semantischer Segmentierung.

[0048] Fig. 4A und Fig. 4B zeigen einen Vision-Sprache-Einbettungsraum 400, der zwei unterschiedliche Mengen von Pixeleinbettungen zusammen mit ihrer Semantik umfasst. Spezieller zeigen Fig. 4A und Fig. 4B eine Menge von Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210, die durch die Kontur 402 umrissen wird, und eine Menge von Pixeleinbettungen des Maschinenlernmodells 220, die durch die Kontur 404 umrissen wird. Außerdem umfasst die Kontur 402 zum Beispiel C_i zum Repräsentieren eines Klassenprototyps und C_j zum Repräsentieren eines anderen Klassenprototyps. Zum Beispiel kann C_i als nicht einschränkendes Beispiel einen Klassenprototyp repräsentieren, der einer Jet-Klasse zugeordnet ist, und C_j kann einen Klassenprototyp repräsentieren, der einer Mond-Klasse zugeordnet ist, wie in anderen nicht einschränkenden Beispielen der vorliegenden Offenbarung besprochen. Zwischenzeitlich umfasst die Kontur 404 auch zum Beispiel C_i zum Repräsentieren eines Klassenprototyps und „unbekannt“ zum Repräsentieren eines anderen Klassenprototyps. In dieser Hinsicht entspricht C_i der Kontur 402, das aus Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 erzeugt wird, und C_i entspricht der Kontur 404, das aus Pixeleinbettungen des Maschinenlernmodells 220 erzeugt wird. Außerdem entspricht C_j der Kontur 402, das aus Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 erzeugt wird, dem „unbekannten“ der Kontur 404, das aus Pixeleinbettungen des Maschinenlernmodells 220 erzeugt wird.

[0049] Fig. 4A zeigt ein nicht einschränkendes Beispiel, in dem eine Kontur 402 einer Menge von Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 nicht mit einer Kontur 404 einer Menge von Pixeleinbettungen des Maschinenlernmodells 220 ausgerichtet ist. Wie in Fig. 4A gezeigt, ist dann, wenn die Kontur 402 nicht mit der Kontur 404 ausgerichtet ist, der Klassenprototyp C_i des vortrainierten Vision-Sprache-Modells 210 im Vision-Sprache-Einbettungsraum 400 nicht mit dem Klassenprototyp C_i des Maschinenlernmodells 220 ausgerichtet. Außerdem ist dann, wenn die Kontur 402 nicht mit der Kontur 404 ausgerichtet ist, der Klassenprototyp C_j des vortrainierten Vision-Sprache-Modells 210 im Vision-Sprache-Einbettungsraum 400 nicht mit dem entsprechenden Klassenprototyp „unbekannt“ des

Maschinenlernmodells 220 ausgerichtet. Zusätzlich zeigen Fig. 4A und Fig. 4B eine Kontur 406 der Texteinbettungen des vortrainierten Vision-Sprache-Modells 210 als getrennt von der Kontur 402 im Vision-Sprache-Einbettungsraum 400. Fig. 4A und Fig. 4B zeigen außerdem die Kontur 406 der Texteinbettungen des vortrainierten Vision-Sprache-Modells 210 als getrennt von der Kontur 404 im Vision-Sprache-Einbettungsraum 400.

[0050] Wie bereits in der vorliegenden Offenbarung erwähnt wurde, verwendet zur Ermöglichung von sprachengeführter semantischer Segmentierung und Verbesserung der Qualität von Pixeleinbettungen der Rahmen 201 ein vortrainiertes Vision-Sprache-Modell 210 (z. B. CLIP) zur Führung des Trainings der Pixeleinbettungsfunktion des Maschinenlernmodells 220. Der Rahmen 200 richtet einen Ausgangsraum von Pixeleinbettungsfunktion des Maschinenlernmodells 220 so aus, dass es mit einem Merkmalraum des vortrainierten Vision-Sprache-Modells 210 vereinbar ist. Der Rahmen 200 ist ausgelegt zum Führen und/oder Ausrichten der Pixeleinbettungen, die aus dem selbstbeaufsichtigten Verfahren des Maschinenlernmodells 220 erzeugt werden, mit den Pixeleinbettungen, die aus dem vortrainierten Vision-Sprache-Modell 210 erzeugt werden. Der Rahmen 200 führt diese Ausrichtung durch Minimieren der Distanz zwischen den Pixeleinbettungen des Maschinenlernmodells 220 und den Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 im Vision-Sprache-Einbettungsraum 400 durch.

[0051] Fig. 4B zeigt ein nicht einschränkendes Beispiel, in dem die Kontur 402 der Pixeleinbettungen des vortrainierten Vision-Sprache-Modells 210 mit der Kontur 404 der Pixeleinbettungen des Maschinenlernmodells 220 im Vision-Sprache-Einbettungsraum 400 über den Rahmen 200 ausgerichtet oder mindestens teilweise ausgerichtet (oder mindestens teilweise überlappend) ist. In diesem nicht einschränkenden Beispiel ist die Kontur 404 von Fig. 4B mindestens aufgrund des durch den Rahmen 200 bereitgestellten Trainings größer als die Kontur 404 von Fig. 4A. Außerdem zeigt Fig. 4B den Klassenprototyp C_i des Maschinenlernmodells 220 als im Vision-Sprache-Einbettungsraum 400 mit dem Klassenprototyp C_i des vortrainierten Vision-Sprache-Modells 210 ausgerichtet oder mindestens teilweise ausgerichtet (oder zumindest teilweise überlappend). Außerdem ist der Klassenprototyp „unbekannt“ des Maschinenlernmodells 220 im Vision-Sprache-Einbettungsraum 400 mit dem Klassenprototyp C_j des vortrainierten Vision-Sprache-Modells 210 ausgerichtet oder mindestens teilweise ausgerichtet (oder zumindest teilweise überlappend). Wie oben mit Bezug auf Fig. 4A und Fig. 4B gezeigt und besprochen, ist der Rahmen 200 insofern vorteilhaft, als er die Pixeleinbettungen und ihre Semantik (z. B.

Kontur 406 von Texteinbettungen) im Vision-Sprache-Einbettungsraum 400 mindestens teilweise ausgerichtet (oder teilweise überlappen lässt), sodass die Einbettungsqualität und das Lernen von Repräsentationen für unbekannte Klassen verbessert werden.

[0052] Fig. 5 ist ein Flussdiagramm 500 eines nicht einschränkenden Beispiels für den Rahmen 200 während Schlussfolgerung. Spezieller ist während Schlussfolgerung der Rahmen 200 ausgelegt zum Empfangen eines Digitalbildes 502 als Eingangsdaten. In diesem nicht einschränkenden Beispiel zeigt das Digitalbild 502 einen Jet 502A, einen Mond 502B und einen Himmel 502C. Auf den Empfang dieses Digitalbildes 502 hin ist das Maschinenlernmodell 220 ausgelegt zum Erzeugen von Pixeleinbettungen als Ausgabe bei Empfang des Digitalbildes 502 als Eingabe. Der Rahmen 200 ist dann ausgelegt zum Erzeugen einer Menge von Segmenteinbettungen, wie in Bild 504 gezeigt, unter Verwendung der Pixeleinbettungen des Maschinenlernmodells 220. Die Menge von Segmenteinbettungen wird auf der Basis von Clustern von Pixeleinbettungen erzeugt, die sich ähnliche semantische Bedeutungen teilen oder zu ähnlichen/denselben visuell kohärenten Regionen gehören. Zum Beispiel umfasst die Menge von Segmenteinbettungen eine Segmenteinbettung 504A für den Jet, eine Segmenteinbettung 504B für den Mond und eine Anzahl von Segmenteinbettungen 504C für den Himmel.

[0053] Zusätzlich kann eine Menge von Klassen 506 (z. B. Jet, Katze, Mond usw.) durch einen Benutzer (z. B. einen Softwareentwickler) über eine Aufforderungsvorlage 304 eingegeben werden. In diesem nicht einschränkenden Beispiel ist die Aufforderungsvorlage 304 „ein Foto einer/eines {Klassenname}“, wobei {Klassenname} die Benutzereingabe (z. B. Klassenname wie etwa Jet) repräsentiert. In diesem nicht einschränkenden Beispiel umfasst die Menge von Klassen 506 zum Beispiel eine Jet-Klasse 506A, eine Mond-Klasse 506B und eine Anzahl anderer Klassen (z. B. Katzenklasse usw.). Außerdem ist, wie in Fig. 5 gezeigt, der modifizierte Bildcodierer 210A (z. B. Mask CLIP) ausgelegt zum Erzeugen einer Menge von Texteinbettungen 508 als Ausgabe bei Empfang der Menge von Klassen 506 als Eingabe. Zum Beispiel ist der modifizierte Bildcodierer 210A ausgelegt zum Erzeugen einer Texteinbettung 508A für die Jet-Klasse 506A, einer Texteinbettung 508B für die Mond-Klasse 506B und eine Anzahl anderer Texteinbettungen für eine Anzahl anderer Klassen.

[0054] Der Rahmen 200 bestimmt dann eine Ähnlichkeit jeder Segmenteinbettung in Bezug auf jede Texteinbettung. Als nichteinschränkendes Beispiel berechnet zum Beispiel der Rahmen 200 eine Kosinusähnlichkeit zwischen der Segmenteinbettung 504A (die dem Jet-Segment entspricht) und der Text-

einbettung 508B (die der Mond-Klasse entspricht). Als ein anderes nichteinschränkendes Beispiel berechnet zum Beispiel der Rahmen 200 eine Kosinusähnlichkeit zwischen der Segmenteinbettung 504B (die dem Mond-Segment entspricht) und der Texteinbettung 508B (die der Mond-Klasse entspricht). Wie in Fig. 5 gezeigt, bestimmt in diesem Fall der Rahmen 200, dass die Segmenteinbettung 504A (die dem Jet-Segment entspricht) eine niedrigere Wahrscheinlichkeit (z. B. Ähnlichkeitsdaten kleiner als 0,5) der Korrelation mit der Texteinbettung 508B (die der Mond-Klasse entspricht) aufweist, während er außerdem bestimmt, dass die Segmenteinbettung 504B eine höhere Wahrscheinlichkeit (z. B. Ähnlichkeitsdaten größer als 0,5) der Korrelation mit der Texteinbettung 508B (die der Mond-Klasse entspricht) aufweist.

[0055] Nach dem Durchführen der Kosinusähnlichkeits-Berechnungen erzeugt der Rahmen 200 semantische Segmentierungsdaten, die mindestens die Vergabe der Menge von Klassen 506 an die Menge von Bildsegmenten (z. B. Maskensegmente), wie in Bild 510 gezeigt, umfasst. Zum Beispiel vergibt der Rahmen 200 die Jet-Klasse 506A an das Bildsegment 510A (und dieses Bildsegment 510A ausmachende Pixel), wenn bestimmt wird, dass die Ähnlichkeitsdaten innerhalb eines vorbestimmten Schwellenbereichs liegen. Der Rahmen 200 vergibt die Mond-Klasse 506B an das Bildsegment 510B (und das Bildsegment 510B ausmachende Pixel), wenn bestimmt wird, dass die Ähnlichkeitsdaten innerhalb eines vorbestimmten Schwellenbereichs liegen. Außerdem vergibt der Rahmen 200 nicht einen Klassennamen aus der Menge von Klassen 506 an das Bildsegment 510C, da die entsprechenden Segmenteinbettungen 504C mit Bezug auf die berechneten Ähnlichkeitsdaten bezüglich des vorbestimmten Schwellenbereichs keine ausreichende Wahrscheinlichkeit/Korrelation mit der Menge von Texteinbettungen 508 aufweisen. In solchen Fällen kann der Rahmen 200 eine Vorgabekennzeichnung (z. B. Hintergrundkennzeichnung) an diese Bildsegmente 510C vergeben.

[0056] In dem Fall, dass der Klassenname „Himmel“ über eine Aufforderungsvorlage 304 während einer späteren Schlussfolgerungszeit eingegeben wird, ist der Rahmen 200 ausgelegt zum Vergeben der Himmel-Klasse an jedes der Anzahl von Bitsegmenten 510C die dem Himmel entsprechen, da der Rahmen 200 diese Segmente während des Trainings als einen unbekannten Klassenprototypen gelernt hat. Der Rahmen 200 ist in der Lage, diese Aktion ähnlich auszuführen, wie der Rahmen 200 in der Lage ist, die Mond-Klasse 506B während Schlussfolgerung an das Bildsegment 510B zu vergeben, obwohl der Mond-Klassenname während des Trainings (Fig. 3) dem Textcodierer 210B nicht als Eingabe bereitgestellt wurde. In dieser Hinsicht ist der Rahmen 200

dahingehend vorteilhaft, dass er ausgelegt ist zum Erzeugen von semantischen Segmentierungsdaten (z. B. einer semantischen Kennzeichnung oder einer Klassenkennzeichnung) für jedes Segment eines Digitalbildes während Schlussfolgerungszeit entweder aus (i) bekannten Klassen, die dem Rahmen 200 (z. B. dem Textcodierer 210B) während des Trainings bereitgestellt wurden, oder (ii) unbekannten Klassen. Die unbekannten Klassen können als Platzhalter für neue Eingangsdaten betrachtet werden, die durch den Rahmen 200 empfangen und transformiert werden können. Zum Beispiel kann ein neuer Klassenname, wie etwa „Himmel“ während Schlussfolgerung an die Menge von Klassen 506 angehängt werden. Dieser neue Klassenname „Himmel“ wird während des Trainings als eine unbekannte Klasse betrachtet und während des Trainings einem bestimmten unbekannten Klassenprototyp zugeordnet. Der Rahmen 200 kann diesen neuen Klassennamen (z. B. Himmel) jedoch während Schlussfolgerungszeit von einem Benutzer empfangen, und der Rahmen 200 wird dazu befähigt, semantische Segmentierungsdaten (z. B. semantische Kennzeichnungen oder Klassenkennzeichnungen) für alle Bildsegmente 510C des Digitalbildes 502, die diesen neuen Eingangsdaten (z. B. dem Klassennamen Himmel) entsprechen, zu erzeugen. In dieser Hinsicht ist der Rahmen 200 während Schlussfolgerungszeit dafür ausgelegt, (i) einem Benutzer zu ermöglichen, einen beliebigen Klassennamen einzugeben, der während des Trainings nicht bekannt ist, und (ii) diesen und mindestens einen beliebigen Klassennamen zu Texteinbettung zu codieren, um mit der ähnlichsten Segmenteinbettung übereinzustimmen.

[0057] Fig. 6 ist eine Darstellung eines Systems 600, das den Rahmen 200 umfasst, der für sprachengeführte selbstbeaufsichtigte semantische Segmentierung ausgelegt ist. In diesem Beispiel umfasst das System 600 mindestens ein Sensorsystem 610, ein Steuersystem 620 und ein Aktorsystem 630. Das System 600 ist so ausgelegt, dass das Steuersystem 620 das Aktorsystem 630 auf der Basis von Sensordaten aus dem Sensorsystem 610 steuert. Spezieller umfasst das Sensorsystem 610 einen oder mehrere Sensoren und/oder entsprechende Vorrichtungen zum Erzeugen von Sensordaten. Zum Beispiel umfasst das Sensorsystem 610 einen Bildsensor, eine Kamera, einen Radarsensor, einen LIDAR-Sensor (Light Detection And Ranging), einen Wärmesensor, einen Ultraschallsensor, einen Infrarotsensor, einen Bewegungssensor, einen Sensor der auf Satelliten basierenden Navigation (z. B. einen GPS-Sensor (Global Positioning System)), einen optischen Sensor, einen Audiosensor, einen beliebigen geeigneten Sensor oder eine beliebige Anzahl und Kombination davon. Nachdem es Detektionen aus der Umgebung erhält, ist das Sensorsystem 610 betreibbar zur Kommunikation mit dem Steuersys-

tem 620 über einen Eingabe-/Ausgabe- bzw. E/A-System 670 und/oder andere Funktionsmodule 650, die Kommunikationstechnologie umfassen.

[0058] Das Steuersystem 620 ist dafür ausgelegt, die Sensordaten direkt oder indirekt von einem oder mehreren Sensoren des Sensorsystems 610 zu erhalten. In dieser Hinsicht können die Sensordaten Sensordaten von einem einzigen Sensor oder Sensorfusionsdaten von mehreren Sensoren umfassen. Auf den Empfang von Eingabe hin, die mindestens Sensordaten umfassen, ist das Steuersystem 620 betreibbar zum Verarbeiten der Sensordaten über das Verarbeitungssystem 640 in dieser Hinsicht umfasst das Verarbeitungssystem 640 mindestens einen Prozessor. Zum Beispiel umfasst das Verarbeitungssystem 640 einen elektronischen Prozessor, eine CPU, eine GPU, einen Mikroprozessor, ein FPGA, ein ASIC, Verarbeitungsschaltungen, beliebige geeignete Verarbeitungstechnologie oder eine beliebige Kombination davon. Auf die Verarbeitung mindestens dieser Sensordaten hin ist das Verarbeitungssystem 640 ausgelegt zum Extrahieren, Erzeugen und/oder Erhalten von ordnungsgemäßen Eingangsdaten (z. B. Digitalbilddaten) für den Rahmen 200. Zusätzlich ist das Verarbeitungssystem 640 betreibbar zum Erzeugen von Ausgangsdaten (z. B. semantischen Segmentierungsdaten mit Bezug auf in Digitalbildern angezeigte Objekte) über den Rahmen 200 auf der Basis von Kommunikation mit dem Speichersystem 660. Zusätzlich ist das Verarbeitungssystem 640 betreibbar zum Bereitstellen von Aktor Steuerdaten für das Aktorsystem 630 auf der Basis der Ausgangsdaten, der semantischen Segmentierungsdaten und/oder Objekterkennungsdaten.

[0059] Das Speichersystem 660 ist ein Computer- oder elektronisches Speichersystem, das dafür ausgelegt ist, verschiedene Daten zu speichern und Zugang zu diesen bereitzustellen, um mindestens die Operationen und Funktionalität wie hier offenbart zu ermöglichen. Das Speichersystem 660 umfasst eine einzige Vorrichtung oder mehrere Vorrichtungen. Das Speichersystem 660 umfasst elektrische, elektronische, magnetische, optische, Halbleiter-, elektromagnetische, beliebige geeignete Speichertechnologie oder eine beliebige Kombination davon. Zum Beispiel kann das Speichersystem 660 RAM, ROM, Flash-Speicher, ein Plattenlaufwerk, eine Speicherkarte, eine optische Speichervorrichtung, eine magnetische Speichervorrichtung, ein Speichermodul, eine beliebige geeignete Art von Speichervorrichtung oder eine beliebige Anzahl und Kombination davon umfassen. Bei einer beispielhaften Ausführungsform ist das Speichersystem 660 mit Bezug auf das Steuersystem 620 und/oder das Verarbeitungssystem 640 lokal, entfernt oder eine Kombination davon (z. B. teilweise lokal und teilweise entfernt). Zum Beispiel kann das Speichersys-

tem 660 mindestens ein Speicherungssystem auf Cloud-Basis (z. B. ein Datenbanksystem auf Cloud-Basis) umfassen, das von dem Verarbeitungssystem 640 und/oder anderen Komponenten des Steuersystems 620 entfernt ist.

[0060] Das Speichersystem 660 umfasst mindestens den Rahmen 200, der über das Verarbeitungssystem 640 ausgeführt wird. Der Rahmen 200 ist ausgelegt zum Empfangen oder Erhalten von Eingangsdaten, die mindestens ein Digitalbild umfassen. Zusätzlich ist der Rahmen 200 über das Verarbeitungssystem 640 ausgelegt zum Erzeugen von Ausgangsdaten (z. B. semantischen Segmentierungsdaten) auf der Basis der Eingangsdaten (z. B. Digitalbild). Wie bereits besprochen, umfasst der Rahmen 200 mindestens das vortrainierte Vision-Sprache-Modell 210 und das Maschinenlernmodell 220. In diesem Beispiel wurde das Maschinenlernmodell 220 wie oben besprochen unter Verwendung der kombinierten Verlustdaten $\mathcal{L}$ und der Prototyp-Verlustdaten $\mathcal{L}_p$ trainiert. Zusätzlich umfasst das Speichersystem 660 eine Computervision-Anwendung 680, die computerlesbare Daten mit Anweisungen umfasst, die die Ausgangsdaten des Rahmens 200 verwenden, um eine Anzahl von Computervision-Diensten für das Steuersystem 620 bereitzustellen. Die Computervision-Anwendung 680 arbeitet mit dem Rahmen 200 zum Bereitstellen einer Anzahl von Computervision-Diensten (z. B. Objekterkennung) für das Steuersystem 620, sodass das Steuersystem 620 das Aktorsystem 630 gemäß den Computervision-Diensten steuern kann. Das Speichersystem 660 ist außerdem ausgelegt zum Speichern anderer relevanter Daten 690, die den Betrieb des Systems 600 in Bezug auf eine oder mehrere Komponenten (z. B. Sensorsystem 610, Aktorsystem 630 usw.) betreffen.

[0061] Ferner umfasst das System 600, wie in **Fig. 6** gezeigt, andere Komponenten, die zum Betrieb des Steuersystems 620 in Bezug auf das Sensorsystem 610 und das Aktorsystem 630 beitragen. Außerdem umfasst, wie in **Fig. 6** gezeigt, das Steuersystem 620 das E/A-System 670, das eine oder mehrere Schnittstellen für eine oder mehrere E/A-Vorrichtungen umfasst, die das System 600 betreffen. Zum Beispiel stellt das E/A-System 670 mindestens eine Schnittstelle für das Sensorsystem 610 und mindestens eine Schnittstelle für das Aktorsystem 630 bereit. Außerdem ist das Steuersystem 620 ausgelegt zum Bereitstellen anderer Funktionsmodule 650, wie etwa beliebige geeignete Hardwaretechnologie, Softwaretechnologie oder eine beliebige Kombination davon, die bei der Funktionsweise des Systems 600 hilft und/oder zu dieser beiträgt. Zum Beispiel umfassen die anderen Funktionsmodule 650 ein Betriebssystem und Kommunikationstechnologie, wodurch Komponenten des Systems 600 wie hier beschrieben miteinander kommunizieren können.

Mit mindestens der in dem Beispiel von **Fig. 6** besprochenen Ausgestaltung ist das System 600 in verschiedenen Technologien anwendbar.

[0062] **Fig. 7** ist eine Darstellung des Systems 600 in Bezug auf mobile Maschinentechнологie 700 gemäß einer beispielhaften Ausführungsform. Als nichteinschränkendes Beispiel umfasst die mobile Maschinentechнологie 700 ein mindestens teilweise autonomes Fahrzeug oder einen mindestens teilweise autonomen Roboter. In **Fig. 7** ist die mobile Maschinentechнологie 700 ein mindestens teilweise autonomes Fahrzeug, das ein Sensorsystem 610 umfasst. Das Sensorsystem 610 umfasst einen optischen Sensor, einen Bildsensor, einen Videosensor, einen Ultraschallsensor, einen Positionssensoren (z. B. GPS-Sensor), einen Radarsensor, einen Lidar-Sensor, einen beliebigen geeigneten Sensor oder eine beliebige Anzahl und Kombination davon. Ein oder mehrere der Sensoren können mit Bezug auf das Fahrzeug integriert sein. Das Sensorsystem 610 ist dafür ausgelegt, dem Steuersystem 620 Sensordaten bereitzustellen.

[0063] Das Steuersystem 620 ist ausgelegt zum Erhalten oder Erzeugen von Bilddaten, die auf Sensordaten oder Sensorfusionsdaten aus dem Sensorsystem 610 basieren. Zusätzlich ist das Steuersystem 620 ausgelegt zum Vorverarbeiten der Sensordaten, um dem Rahmen 200 Eingangsdaten einer geeigneten Form (z. B. Digitalbilddaten) bereitzustellen. Der Rahmen 200 ist vorteilhafterweise ausgelegt zum Bereitstellen von semantischen Segmentierungsdaten, sodass in den Sensordaten angezeigte Objekte detektiert und erkannt werden können.

[0064] Zusätzlich ist das Steuersystem 620 ausgelegt zum Erzeugen von Aktorstuerdaten, die mindestens auf Ausgangsdaten (z. B. semantischen Segmentierungsdaten, Objektidentifikationsdaten usw.) des Rahmens 200 gemäß der Computervision-Anwendung 680 basieren. In dieser Hinsicht ist das Steuersystem 620 ausgelegt zum Erzeugen von Aktorstuerdaten, die sicherere und genauere Steuerung des Aktorsystem 630 des Fahrzeugs durch die durch den Rahmen 200 bereitgestellte verbesserte semantische Segmentierung erlauben. Das Aktorsystem 630 kann ein Bremssystem, ein Antriebssystem, ein Triebwerk, einen Antriebsstrang, ein Lenksystem oder eine beliebige Anzahl und Kombination von Aktoren des Fahrzeugs umfassen. Das Aktorsystem 630 ist ausgelegt zum Steuern des Fahrzeugs dergestalt, dass das Fahrzeug auf der Basis der Ausgangsdaten (z. B. semantischen Segmentierungsdaten), die durch den Rahmen 200 als Reaktion auf Empfang eines oder mehrerer Digitalbilder auf der Basis der Sensordaten erzeugt werden, Straßenverkehrsregeln befolgt und Kollisionen vermeidet.

[0065] Fig. 8 ist eine Darstellung des Systems 600 mit Bezug auf Sicherheitstechnologie 800 gemäß einer beispielhaften Ausführungsform. Als nichteinschränkendes Beispiel umfasst die Sicherheitstechnologie 800 mindestens ein Überwachungssystem, ein Kontroll-Zugangssystem, ein Beaufsichtigungssystem oder eine beliebige geeignete Art von Sicherheitsvorrichtung. Zum Beispiel betrifft als ein Beispiel Fig. 8 Sicherheitstechnologie 800, die dafür ausgelegt ist, physisch einen Verriegelt-Zustand und einen und Unverriegelt-Zustand eines Schlosses der Tür 802 zu steuern und ein erweitertes Bild/Video auf der Anzeigetechnologie 804 anzuzeigen. Die Sicherheitstechnologie 800 kann auch einen Alarm auslösen und/oder anderen Kommunikationsvorrichtungen/-technologien elektronische Benachrichtigungen bereitstellen. In diesem Beispiel umfasst das Sensorsystem 610 mindestens einen Bildsensor, der dafür ausgelegt ist, Bild-/Videodaten bereitzustellen. Das Sensorsystem 610 kann auch andere Sensoren umfassen, wie etwa einen Bewegungssensor, einen Infrarotsensor usw.

[0066] Das Steuersystem 620 ist ausgelegt zum Erhalten der Bild-/Videodaten von dem Sensorsystem 610. Das Steuersystem 620 ist außerdem ausgelegt zum Erzeugen von semantischen Segmentierungsdaten über den Rahmen 200 auf der Basis der von dem Sensorsystem 610 erhaltenen Bild-/Videodaten. Zusätzlich ist das Steuersystem 620 ausgelegt zum Erzeugen von Aktorsteuerdaten, die sicherere und genauere Steuerung des Aktorsystems 630 erlaubt, indem Ausgangsdaten (z. B. semantischen Segmentierungsdaten) verwendet werden, die durch den Rahmen 200 erzeugt werden. Das Steuersystem 620 ist ausgelegt zum Anzeigen der Ausgangsdaten des Rahmens 200 oder von beliebigen Daten in Bezug auf die Computervision-Anwendung 680 oder eine beliebige Anzahl und Kombination davon auf der Anzeigetechnologie 804.

[0067] Fig. 9 ist eine Darstellung des System 600 mit Bezug auf Bildgebungstechnologie 900 gemäß einer beispielhaften Ausführungsform. Als nichteinschränkendes Beispiel umfasst die Bildgebungstechnologie 900 eine MRI-Vorrichtung (Magnetresonanztomographie), eine Röntgen-Bildgebungsvorrichtung, eine Ultraschallvorrichtung, eine medizinische Bildgebungsvorrichtung, eine beliebige geeignete Art von Bildgebungsvorrichtung oder eine beliebige Anzahl und Kombination davon. In Fig. 9 umfasst das Sensorsystem 610 mindestens einen Bildsensor. Das Steuersystem 620 ist ausgelegt zum Erhalten von Bilddaten von dem Sensorsystem 610. Außerdem ist das Steuersystem 620 über den Rahmen 200 ausgelegt zum Erzeugen von semantischen Segmentierungsdaten auf der Basis der Sensordaten (z. B. Bilddaten). Außerdem ist das Steuersystem 620 ausgelegt zum Bereitstellen von semantischen Segmentierungsdaten und

Objektdetektions-/ -erkennungsdaten mit Bezug auf die Bilddaten des Sensorsystems 610. Zusätzlich ist das Steuersystem 620 ausgelegt zum Anzeigen der beliebigen relevanten Daten (z. B. Sensordaten, Ausgangsdaten des Rahmens 200, relevanten Daten aus der Computervision-Anwendung 680 oder einer beliebigen Anzahl und Kombination davon) auf der Anzeige 902.

[0068] Wie oben beschrieben, sind die Ausführungsformen vorteilhaft und stellen eine Anzahl von Nutzen bereit. Zum Beispiel stellt der Rahmen 200 einen neuartigen Pixelrepräsentations-Lernansatz für semantische Segmentierung bereit. Der Rahmen 200 kombiniert das selbstbeaufsichtigte kontrastive Lernen des Maschinenlernmodells 220 mit der Führung des vortrainierten Vision-Sprache-Modells 210 (z. B. CLIP), um einheitliche Pixeleinbettungen mit Bezug auf visuelle und konzeptuelle Semantik zu lernen. Der Rahmen 200 erweitert selbstbeaufsichtigtes kontrastives Lernen von Bildern auf Pixel, indem jede Pixeleinbettung zu ihren positiven Paaren angezogen und von ihren negativen Paaren abgestoßen wird.

[0069] Außerdem ist der Rahmen 200 vorteilhaft bei der Ermöglichung einer neuen Aufgabe, nämlich klassenfreie semantischen Segmentierung. Diese neue Aufgabe benötigt keinerlei menschliche Annotationen und nimmt sogar nicht an, dass während des Trainings Klassennamen gegeben sind. Dies ist eine problematischere Aufgabe als die jüngste Arbeit (z. B. MaskCLIP+), die Klassennamen sowohl bekannter Klassen als auch unbekannter Klassen erfordert. Zum Beispiel nimmt verglichen mit MaskCLIP+ mit Bezug auf die neue Aufgabe der Rahmen 200 an, dass während des Trainings keine unbekannte Klassennamen gegeben sind. Außerdem werden die Pixeleinbettungen von MaskCLIP+ durch die vordefinierten Klassen verzerrt, wodurch die Fähigkeit von MaskCLIP+, Unbekannte zu erkennen, begrenzt wird. Außerdem benötigt MaskCLIP+ während des Trainings unbekannte Klasseninformationen, wodurch Anwendungen in der realen Welt behindert werden.

[0070] Zusätzlich zeigt der Rahmen 200 einheitliche und beträchtliche Gewinne gegenüber anderen Verfahren der unbeaufsichtigten und sprachengeführten semantischen Segmentierung an mindestens vier Datenmengen: Pascal VOC 2012, Pascal Context und COCO-Stuff und DAVIS 2017. Außerdem übertrifft der Rahmen 200 signifikant die modernste Technik mit Bezug auf die Segmentierung unbekannter Klassen. Darüber hinaus ist der Rahmen 200 dafür ausgelegt, unbekannte Klassen während Schlussfolgerungszeit zu erkennen, ohne während Trainingszeit irgendwelche unbekannten Klasseninformationen zu kennen, und lernt auch feinkörnigere und

schärfere Pixeleinbettungen mit Selbstbeaufsichtigung.

[0071] Das heißt, die obige Beschreibung soll nicht einschränkend, sondern veranschaulichend sein und wird im Kontext einer bestimmten Anwendung und ihrer Anforderungen bereitgestellt. Für Fachleute ist aus der obigen Beschreibung erkennbar, dass die vorliegende Erfindung in vielfältigen Formen implementiert werden kann und dass die verschiedenen Ausführungsformen alleine oder in Kombination implementiert werden können. Obwohl Ausführungsformen der vorliegenden Erfindung in Verbindung mit bestimmten Beispielen für diese beschrieben wurden, können deshalb die hier definierten allgemeinen Prinzipien auf andere Ausführungsformen und Anwendungen angewandt werden, ohne vom Gedanken und Schutzzumfang der beschriebenen Ausführungsformen abzuweichen, und der wahre Schutzzumfang der Ausführungsformen und/oder Verfahren der vorliegenden Erfindung ist nicht auf die gezeigten und beschriebenen Ausführungsformen beschränkt, da dem erfahrenen Praktiker bei Durchsicht der Zeichnungen, der Beschreibung und der folgenden Ansprüche verschiedene Modifikationen ersichtlich werden. Zusätzlich oder als Alternative können Komponenten und Funktionalität anders aufgetrennt oder kombiniert werden, als auf die Weise der verschiedenen beschriebenen Ausführungsformen, und können unter Verwendung anderer Terminologie beschrieben werden. Diese und andere Varianten, Modifikationen, Zusätze und Verbesserungen können in den Schutzzumfang der Offenbarung, so wie er in den folgenden Ansprüchen definiert wird, fallen.

Patentansprüche

1. Computerimplementiertes Verfahren zur sprachengeführten selbstbeaufsichtigten semantischen Segmentierung, wobei das Verfahren Folgendes umfasst:

Empfangen eines Quellenbildes;

Erzeugen eines modifizierten Bildes durch Ausführen von Datenergänzung an dem Quellenbild, wobei das modifizierte Bild eine modifizierte Version des Quellenbildes ist;

Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell;

Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen;

Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell;

Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen aus-

geführt wird;

Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen;

Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen; Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten; und

Bereitstellen eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische Segmentierung, wobei der Rahmen (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden, umfasst.

2. Computerimplementiertes Verfahren nach Anspruch 1, ferner umfassend:

Erzeugen eines weiteren modifizierten Bildes durch Ausführen einer weiteren Datensegmentierung an dem Quellenbild, wobei das weitere modifizierte Bild eine weitere modifizierte Version des Quellenbildes ist;

Erzeugen von dritten Pixeleinbettungen auf der Basis des weiteren modifizierten Bildes über das Maschinenlernmodell;

Erzeugen von dritten Segmenteinbettungen unter Verwendung der dritten Pixeleinbettungen; und

Erzeugen von kontrastiven Verlustdaten durch Berechnen von Ähnlichkeitsdaten der Pixeleinbettungen mit Bezug auf die ersten Segmenteinbettungen und die dritten Segmenteinbettungen, wobei die kombinierten Verlustdaten ferner die kontrastiven Verlustdaten umfassen.

3. Computerimplementiertes Verfahren nach Anspruch 1, ferner umfassend:

Einrichten einer Menge von Klassenprototypen, die mindestens (i) einen ersten Klassenprototyp, der einer vorbestimmten Klasse zugeordnet ist, in der dem Rahmen während des Trainings ein Klassenname gegeben wird, und (ii) einen zweiten Klassenprototyp, der einer Platzhalterklasse zugeordnet ist, umfasst;

Erzeugen einer Pseudokennzeichnung für eine bestimmte zweite Segmenteinbettung der zweiten Segmenteinbettungen, wobei die Pseudokennzeichnung durch einen bestimmten Klassenprototyp definiert wird, der eine größte Ähnlichkeit mit der bestimmten zweiten Segmenteinbettung aufweist, wobei der bestimmte Klassenprototyp in der Menge von Klassenprototypen liegt;

Erzeugen einer Klassenvorhersage für eine bestimmte erste Segmenteinbettung, wobei die Klassenvorhersage aus der Menge von Klassenprototypen auf der Basis von Messung größter Ähnlichkeit ausgewählt wird; und

Erzeugen von semantikeinheitlichen Verlustdaten durch Berechnen einer Kreuzentropie der Klassenvorhersage relativ zu der Pseudokennzeichnung,

wobei die kombinierten Verlustdaten ferner die semantikeinheitlichen Verlustdaten umfassen.

4. Computerimplementiertes Verfahren nach Anspruch 3, wobei der erste Klassenprototyp ein Mittelwert einer Ansammlung verschiedener Segmenteinbettungen ist, von denen bestimmt wird, dass sie Texteinbettung des Klassennamens am ähnlichsten sind; und der zweite Klassenprototyp (i) durch eine zufällige Segmenteinbettung initialisiert wird, die aus den zweiten Segmenteinbettungen ausgewählt wird und nicht der vorbestimmten Klasse zugeordnet ist, und (ii) durch Minimieren von Distanzdaten zwischen der zufälligen Segmenteinbettung und einer Gruppe von anderen Segmenteinbettungen, die als zu der Platzhalterklasse gehörend klassifiziert werden, aktualisiert wird.

5. Computerimplementiertes Verfahren nach Anspruch 1, wobei das vortrainierte Vision-Sprache-Modell ein anderes Maschinenlernmodell umfasst, das trainiert wird, Bildklassifikation mit offenen Vokabularklassen durchzuführen; und das Maschinenlernmodell mindestens einen Codierer umfasst.

6. Computerimplementiertes Verfahren nach Anspruch 1, wobei jede erste Segmenteinbettung durch Berechnen eines Mittelwerts eines Clusters der ersten Pixeleinbettungen, die einer selben visuell kohärenten Region zugeordnet sind, erzeugt wird.

7. Computerimplementiertes Verfahren nach Anspruch 1, ferner umfassend: Empfangen eines Digitalbildes; Verwenden des Rahmens in einer Computervision-Anwendung zum Erkennen von Objekten, die in dem Digitalbild angezeigt werden; und Steuern eines Aktors auf der Basis mindestens der erkannten Objekte.

8. System zur sprachengeführten selbstbeaufsichtigten semantischen Segmentierung, wobei das System Folgendes umfasst: einen oder mehrere Prozessoren; mindestens ein nichtflüchtiges computerlesbares Medium in Datenkommunikation mit dem einen oder den mehreren Prozessoren, wobei das mindestens eine nichtflüchtige computerlesbare Medium computerlesbare Daten mit darauf gespeicherten Anweisungen aufweist, die, wenn sie durch den einen oder die mehreren Prozessoren ausgeführt werden, dafür ausgelegt sind, zu bewirken, dass der eine oder die mehreren Prozessoren ein Verfahren ausführen, das Folgendes umfasst: Empfangen eines Quellenbildes; Erzeugen eines modifizierten Bildes durch Ausfüh-

ren von Datenergänzung an dem Quellenbild, wobei das modifizierte Bild eine modifizierte Version des Quellenbildes ist;

Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell;

Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen;

Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell;

Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird;

Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen;

Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen; Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten; und

Bereitstellen eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische Segmentierung, wobei der Rahmen (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden, umfasst.

9. System nach Anspruch 8, wobei das Verfahren ferner Folgendes umfasst:

Erzeugen eines weiteren modifizierten Bildes durch Ausführen einer weiteren Datensegmentierung an dem Quellenbild, wobei das weitere modifizierte Bild eine weitere modifizierte Version des Quellenbildes ist;

Erzeugen von dritten Pixeleinbettungen auf der Basis des weiteren modifizierten Bildes über das Maschinenlernmodell;

Erzeugen von dritten Segmenteinbettungen unter Verwendung der dritten Pixeleinbettungen; und

Erzeugen von kontrastiven Verlustdaten durch Berechnen von Ähnlichkeitsdaten der ersten Pixeleinbettungen mit Bezug auf die ersten Segmenteinbettungen und die dritten Segmenteinbettungen, wobei die kombinierten Verlustdaten ferner die kontrastiven Verlustdaten umfassen.

10. System nach Anspruch 8, wobei das Verfahren ferner Folgendes umfasst:

Einrichten einer Menge von Klassenprototypen, die mindestens (i) einen ersten Klassenprototyp, der einer vorbestimmten Klasse zugeordnet ist, in der dem Rahmen während des Trainings ein Klassenname gegeben wird, und (ii) einen zweiten Klassenprototyp, der einer Platzhalterklasse zugeordnet ist, umfasst;

Erzeugen einer Pseudokennzeichnung für eine bestimmte zweite Segmenteinbettung der zweiten Segmenteinbettungen, wobei die Pseudokennzeichnung durch einen bestimmten Klassenprototyp definiert wird, der eine größte Ähnlichkeit mit der bestimmten zweiten Segmenteinbettung aufweist, wobei der bestimmte Klassenprototyp in der Menge von Klassenprototypen liegt;
 Erzeugen einer Klassenvorhersage für eine bestimmte erste Segmenteinbettung, wobei die Klassenvorhersage aus der Menge von Klassenprototypen auf der Basis von Messung größter Ähnlichkeit ausgewählt wird; und
 Erzeugen von semantikeinheitlichen Verlustdaten durch Berechnen einer Kreuzentropie der Klassenvorhersage relativ zu der Pseudokennzeichnung, wobei die kombinierten Verlustdaten ferner die semantikeinheitlichen Verlustdaten umfassen.

11. System nach Anspruch 10, wobei:
 der erste Klassenprototyp ein Mittelwert einer Ansammlung verschiedener Segmenteinbettungen ist, von denen bestimmt wird, dass sie Texteinbettung des Klassennamens am ähnlichsten sind; und
 der zweite Klassenprototyp (i) durch eine zufällige Segmenteinbettung initialisiert wird, die aus den zweiten Segmenteinbettungen ausgewählt wird und nicht der vorbestimmten Klasse zugeordnet ist, und
 (ii) durch Minimieren von Distanzdaten zwischen der zufälligen Segmenteinbettung und einer Gruppe von anderen Segmenteinbettungen, die als zu der Platzhalterklasse gehörend klassifiziert werden, aktualisiert wird.

12. System nach Anspruch 8, wobei:
 das vortrainierte Vision-Sprache-Modell ein anderes Maschinenlernmodell umfasst, das trainiert wird, Bildklassifikation mit offenen Vokabularklassen durchzuführen; und
 das Maschinenlernmodell mindestens einen Codierer umfasst.

13. System nach Anspruch 8, wobei jede erste Segmenteinbettung durch Berechnen eines Mittelwerts eines Clusters der ersten Pixeleinbettungen, die einer selben visuell kohärenten Region zugeordnet sind, erzeugt wird.

14. System nach Anspruch 8, ferner umfassend:
 Empfangen eines Digitalbildes;
 Verwenden des Rahmens in einer Computervision-Anwendung zum Erkennen von Objekten, die in dem Digitalbild angezeigt werden; und
 Steuern eines Aktors auf der Basis mindestens der erkannten Objekte.

15. Nichtflüchtiges computerlesbares Medium, das computerlesbare Daten mit darauf gespeicherten Anweisungen aufweist, wobei die computerlesbaren Daten durch einen oder mehrere Prozessoren

ausführbar sind, um ein Verfahren auszuführen, das Folgendes umfasst:

Empfangen eines Quellenbildes;
 Erzeugen eines modifizierten Bildes durch Ausführen von Datenergänzung an dem Quellenbild, wobei das modifizierte Bild eine modifizierte Version des Quellenbildes ist;
 Erzeugen von ersten Pixeleinbettungen auf der Basis des modifizierten Bildes über ein Maschinenlernmodell;
 Erzeugen von ersten Segmenteinbettungen unter Verwendung der ersten Pixeleinbettungen;
 Erzeugen von zweiten Pixeleinbettungen auf der Basis des Quellenbildes über ein vortrainiertes Vision-Sprache-Modell;
 Erzeugen von zweiten Segmenteinbettungen durch Anwenden von Segmentkonturdaten der ersten Segmenteinbettungen auf entsprechende Cluster der zweiten Pixeleinbettungen, nachdem die Datenergänzung an den zweiten Pixeleinbettungen ausgeführt wird;
 Erzeugen von einbettungseinheitlichen Verlustdaten mindestens durch Vergleichen der ersten Segmenteinbettungen mit den zweiten Segmenteinbettungen;
 Erzeugen von kombinierten Verlustdaten, die die einbettungseinheitlichen Verlustdaten umfassen;
 Aktualisieren von Parametern des Maschinenlernmodells auf der Basis der kombinierten Verlustdaten; und
 Bereitstellen eines Rahmens für sprachengeführte selbstbeaufsichtigte semantische Segmentierung, wobei der Rahmen (i) das vortrainierte Vision-Sprache-Modell und (ii) das Maschinenlernmodell, nachdem die Parameter aktualisiert wurden, umfasst.

16. Nichtflüchtiges computerlesbares Medium nach Anspruch 15, wobei das Verfahren ferner Folgendes umfasst:

Erzeugen eines weiteren modifizierten Bildes durch Ausführen einer weiteren Datensegmentierung an dem Quellenbild, wobei das weitere modifizierte Bild eine weitere modifizierte Version des Quellenbildes ist;
 Erzeugen von dritten Pixeleinbettungen auf der Basis des weiteren modifizierten Bildes über das Maschinenlernmodell;
 Erzeugen von dritten Segmenteinbettungen unter Verwendung der dritten Pixeleinbettungen; und
 Erzeugen von kontrastiven Verlustdaten durch Berechnen von Ähnlichkeitsdaten der Pixeleinbettungen mit Bezug auf die ersten Segmenteinbettungen und die dritten Segmenteinbettungen, wobei die kombinierten Verlustdaten ferner die kontrastiven Verlustdaten umfassen.

17. Nichtflüchtiges computerlesbares Medium nach Anspruch 15, wobei das Verfahren ferner Folgendes umfasst:

Einrichten einer Menge von Klassenprototypen, die

mindestens (i) einen ersten Klassenprototyp, der einer vorbestimmten Klasse zugeordnet ist, in der dem Rahmen während des Trainings ein Klassenname gegeben wird, und (ii) einen zweiten Klassenprototyp, der einer Platzhalterklasse zugeordnet ist, umfasst;

Erzeugen einer Pseudokennzeichnung für eine bestimmte zweite Segmenteinbettung der zweiten Segmenteinbettungen, wobei die Pseudokennzeichnung durch einen bestimmten Klassenprototyp definiert wird, der eine größte Ähnlichkeit mit der bestimmten zweiten Segmenteinbettung aufweist, wobei der bestimmte Klassenprototyp in der Menge von Klassenprototypen liegt;

Erzeugen einer Klassenvorhersage für eine bestimmte erste Segmenteinbettung, wobei die Klassenvorhersage aus der Menge von Klassenprototypen auf der Basis von Messung größter Ähnlichkeit ausgewählt wird; und

Erzeugen von semantikeinheitlichen Verlustdaten durch Berechnen einer Kreuzentropie der Klassenvorhersage relativ zu der Pseudokennzeichnung, wobei die kombinierten Verlustdaten ferner die semantikeinheitlichen Verlustdaten umfassen.

18. Nichtflüchtiges computerlesbares Medium nach Anspruch 17, wobei:

der erste Klassenprototyp ein Mittelwert einer Ansammlung verschiedener Segmenteinbettungen ist, von denen bestimmt wird, dass sie Texteinbettung des Klassennamens am ähnlichsten sind; und der zweite Klassenprototyp (i) durch eine zufällige Segmenteinbettung initialisiert wird, die aus den zweiten Segmenteinbettungen ausgewählt wird und nicht der vorbestimmten Klasse zugeordnet ist, und (ii) durch Minimieren von Distanzdaten zwischen der zufälligen Segmenteinbettung und einer Gruppe von anderen Segmenteinbettungen, die als zu der Platzhalterklasse gehörend klassifiziert werden, aktualisiert wird.

19. Nichtflüchtiges computerlesbares Medium nach Anspruch 15, wobei:

das vortrainierte Vision-Sprache-Modell ein anderes Maschinenlernmodell umfasst, das trainiert wird, Bildklassifikation mit offenen Vokabularklassen durchzuführen; und

das Maschinenlernmodell mindestens einen Codierer umfasst.

20. Nichtflüchtiges computerlesbares Medium nach Anspruch 15, wobei jede erste Segmenteinbettung durch Berechnen eines Mittelwerts eines Clusters der ersten Pixeleinbettungen, die einer selben visuell kohärenten Region zugeordnet sind, erzeugt wird.

Es folgen 10 Seiten Zeichnungen

Anhängende Zeichnungen

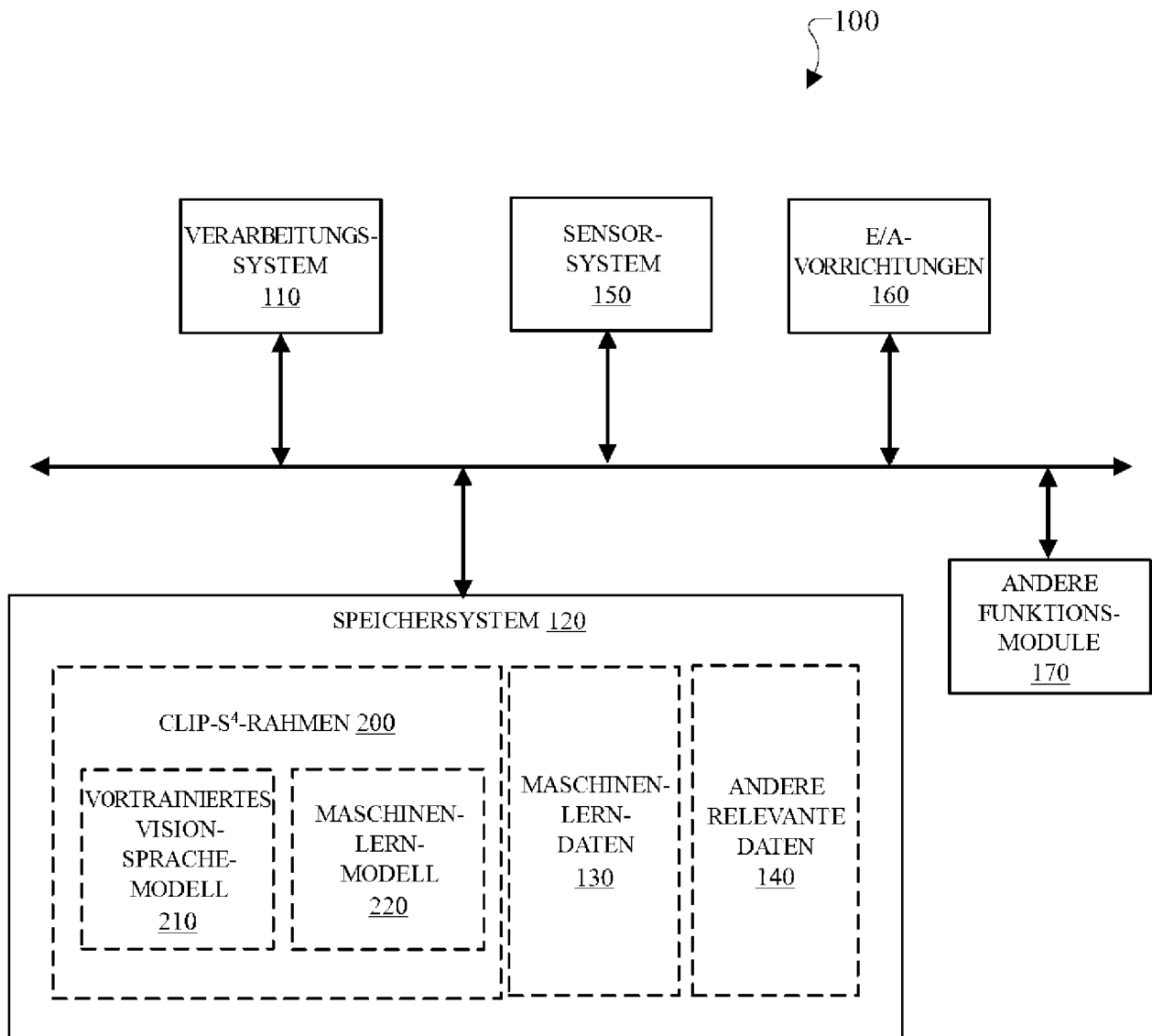


FIG. 1

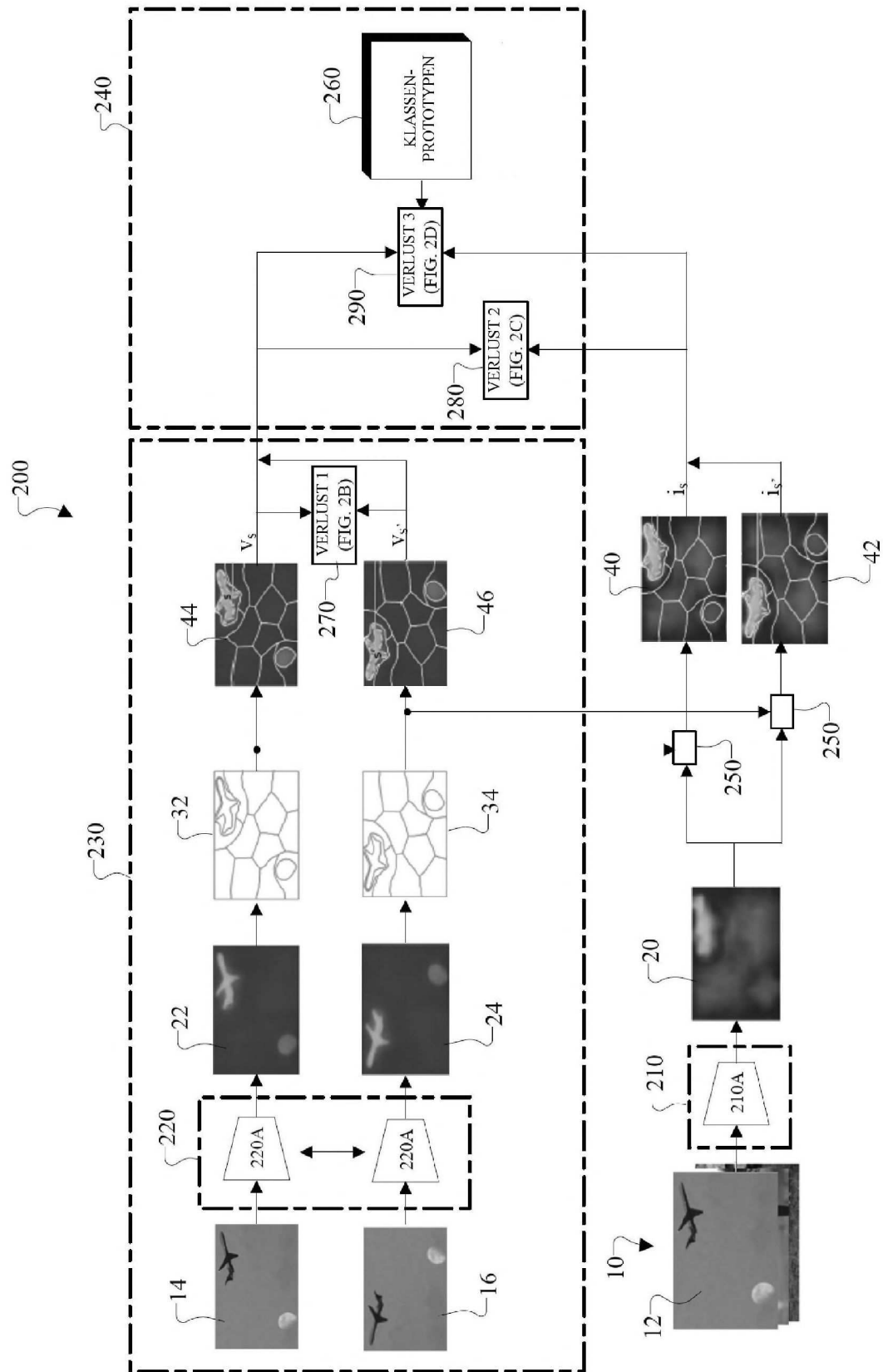


FIG. 2A

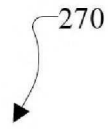


FIG. 2B

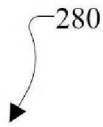


FIG. 2C

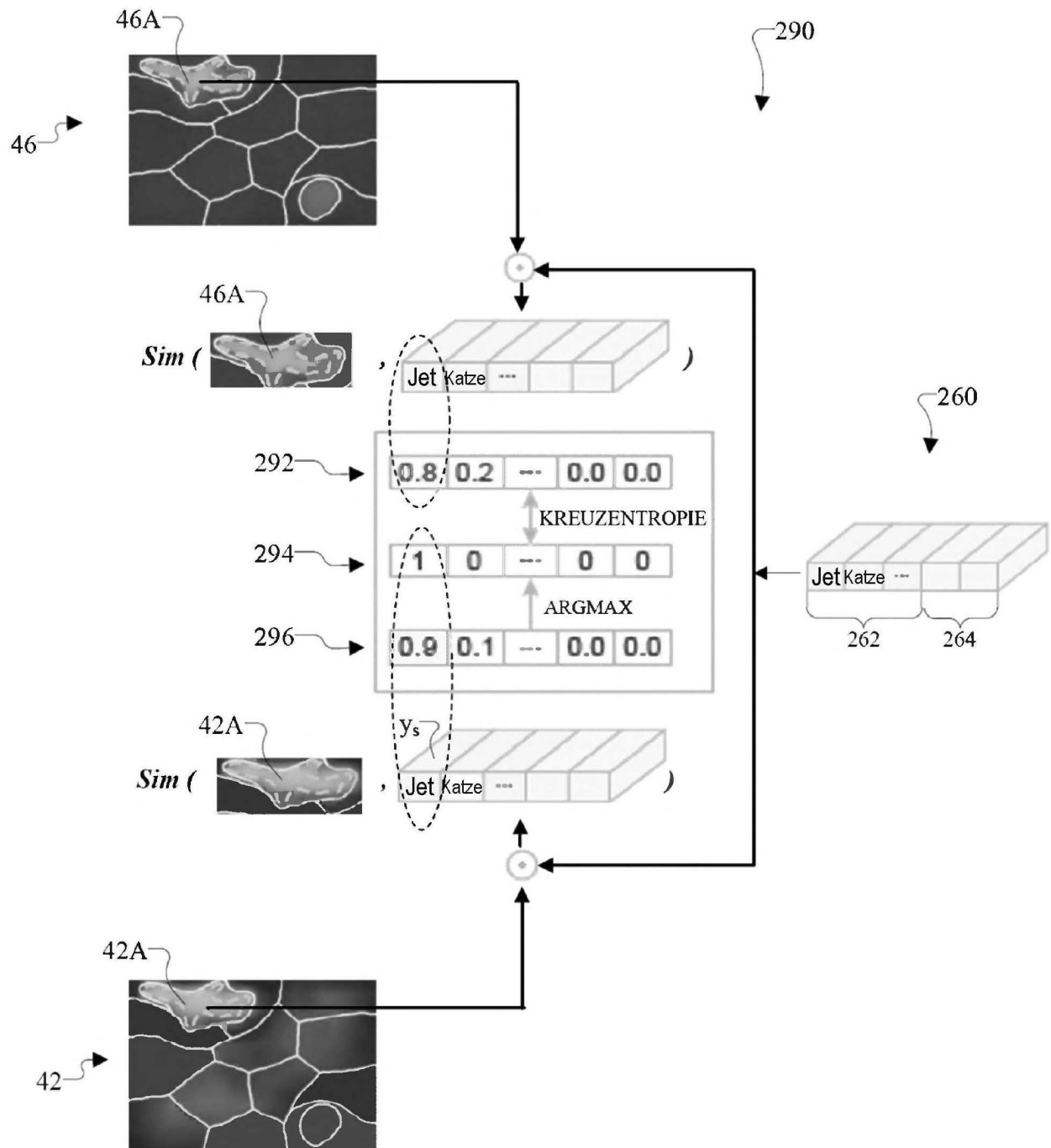


FIG. 2D

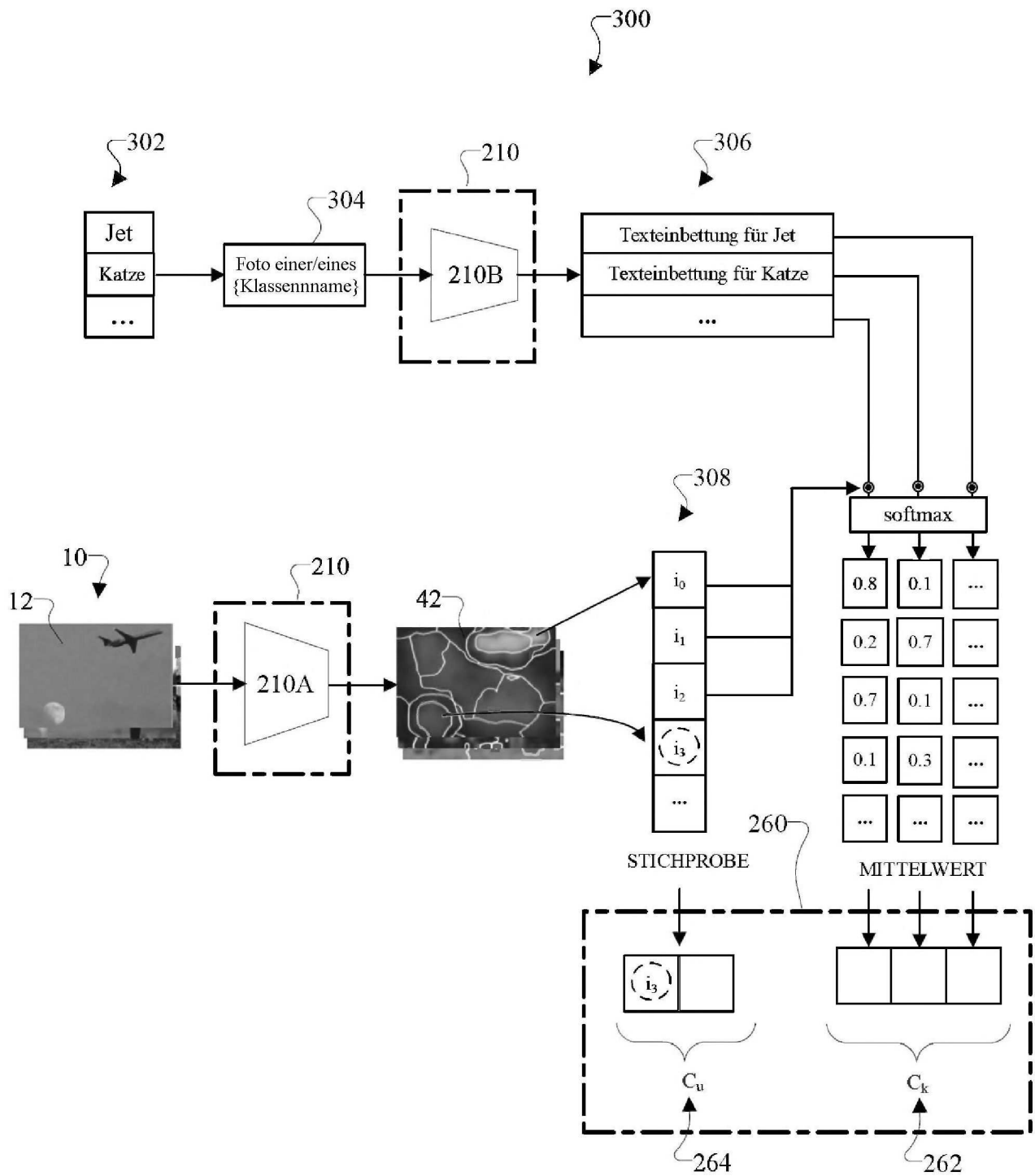


FIG. 3

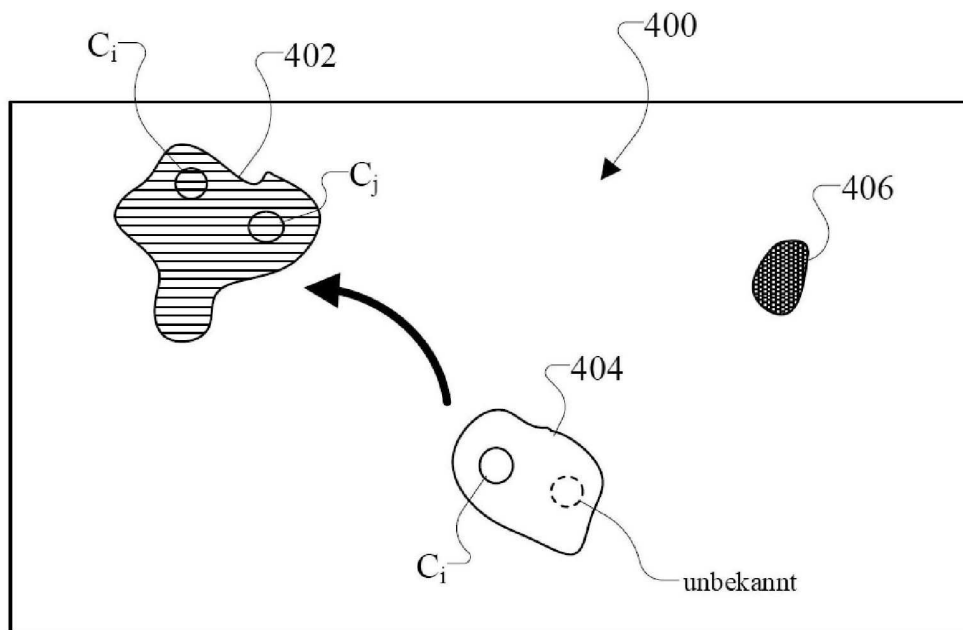


FIG. 4A

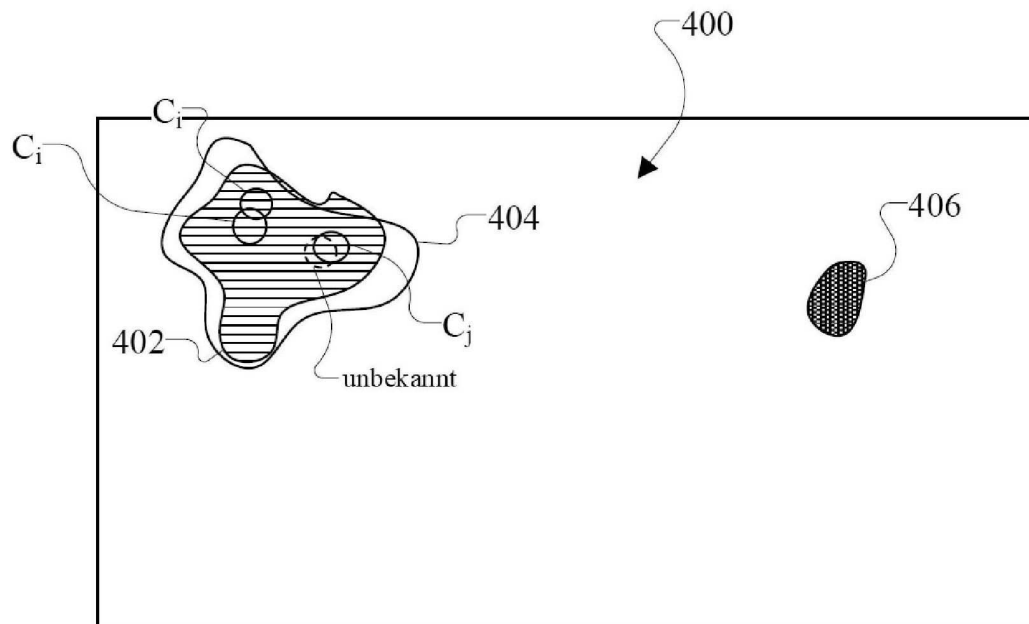


FIG. 4B

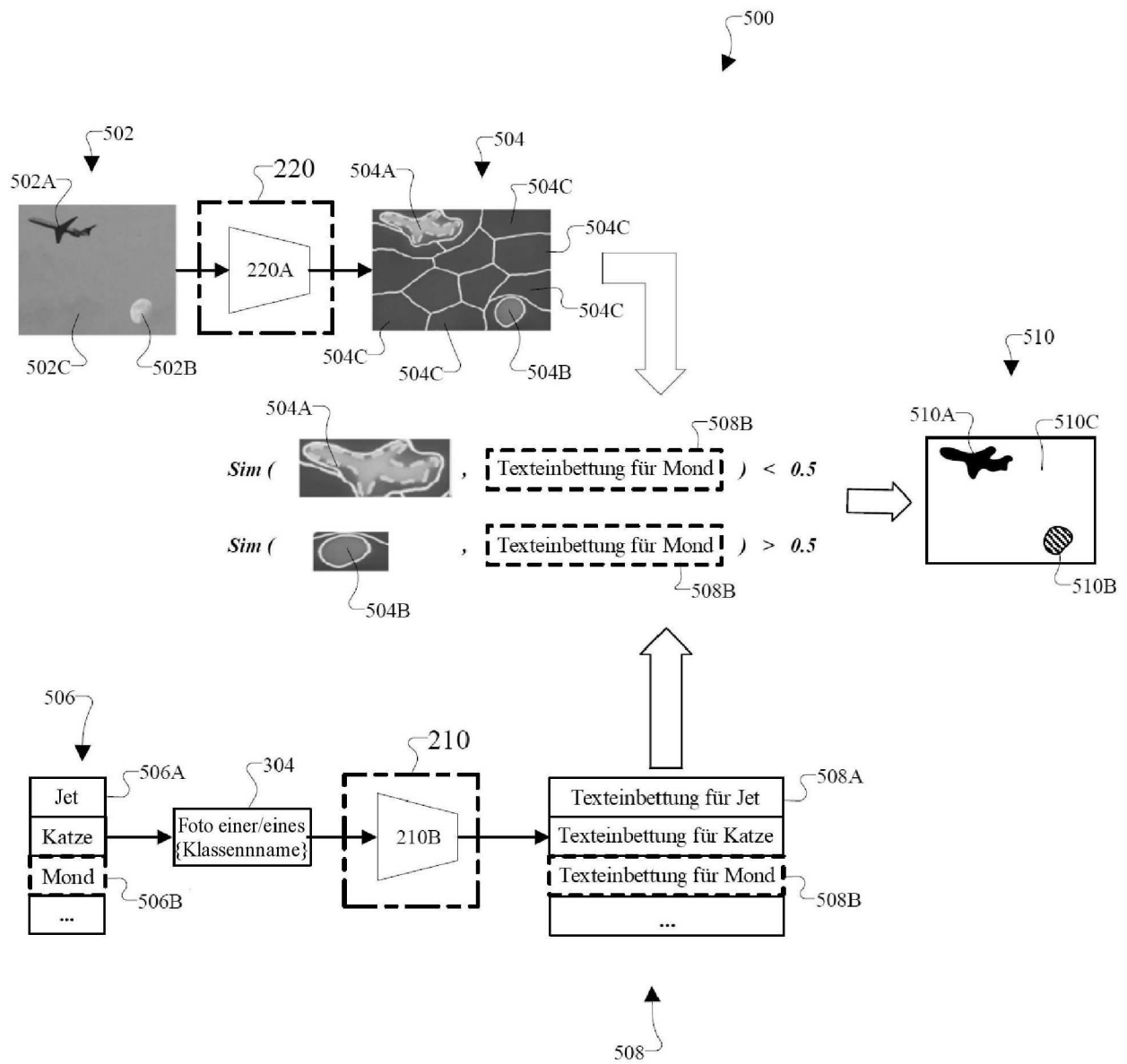


FIG. 5

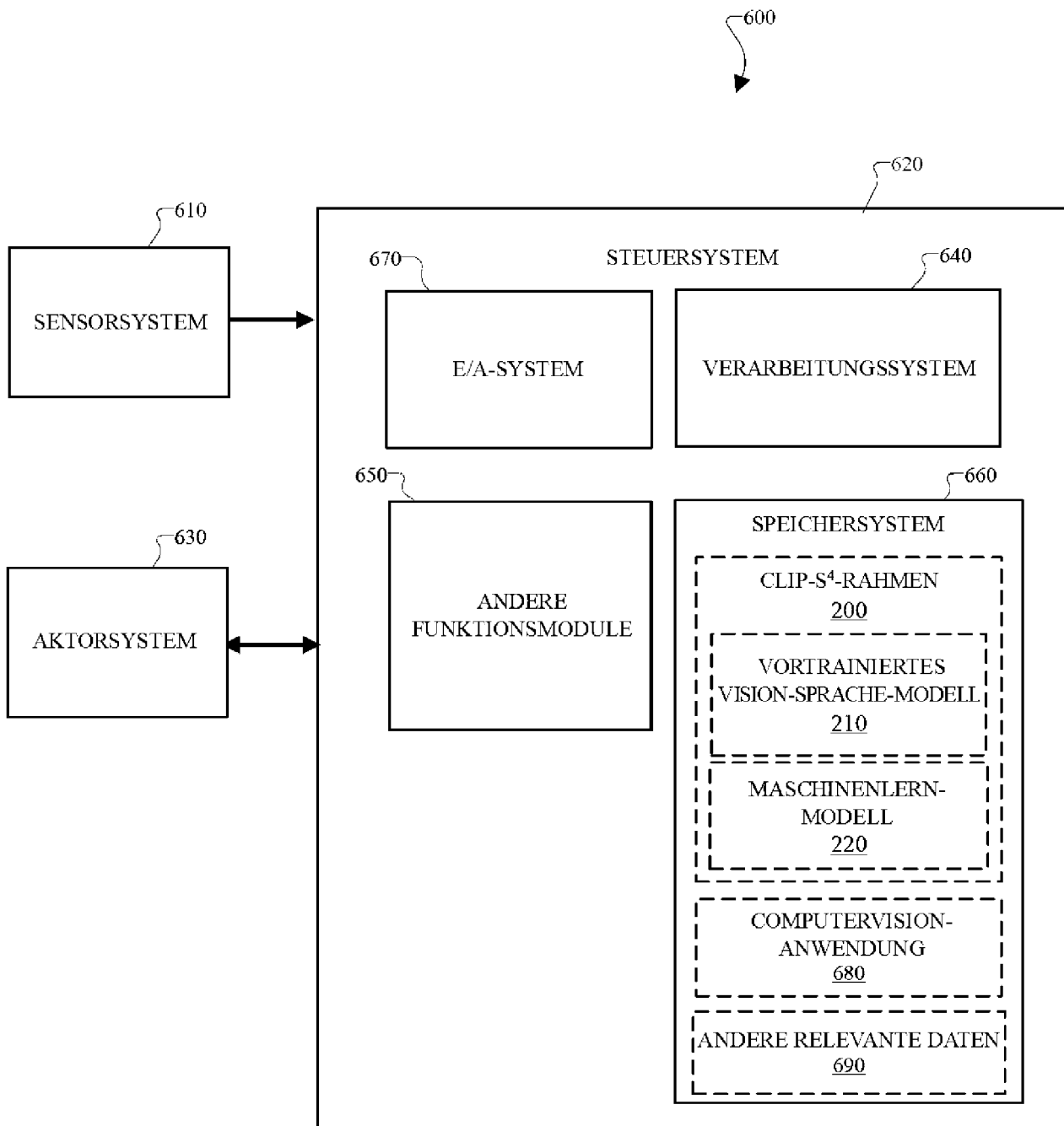


FIG. 6

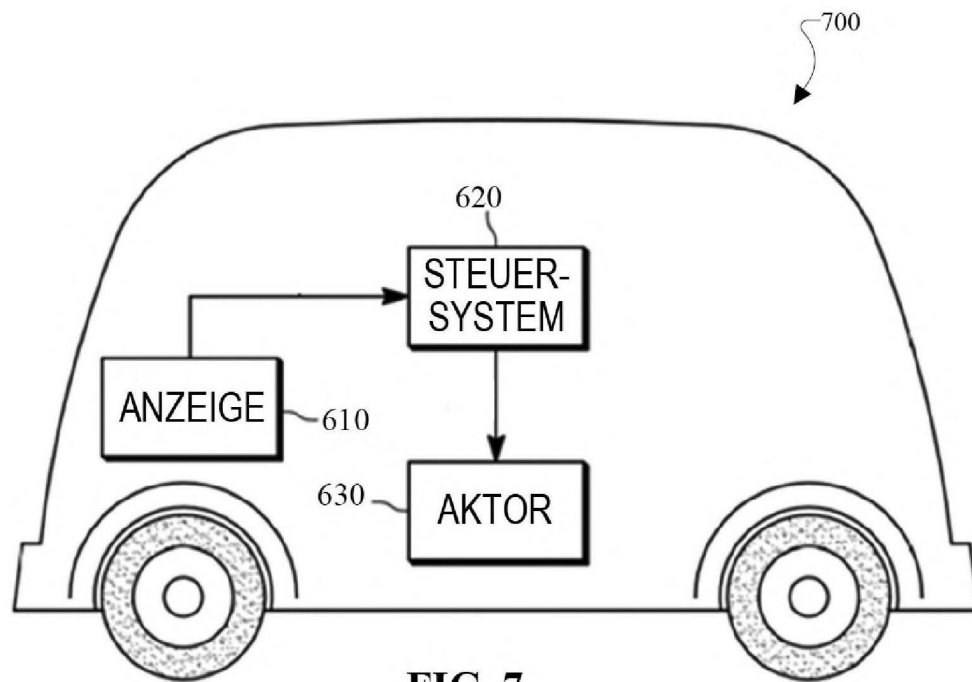


FIG. 7

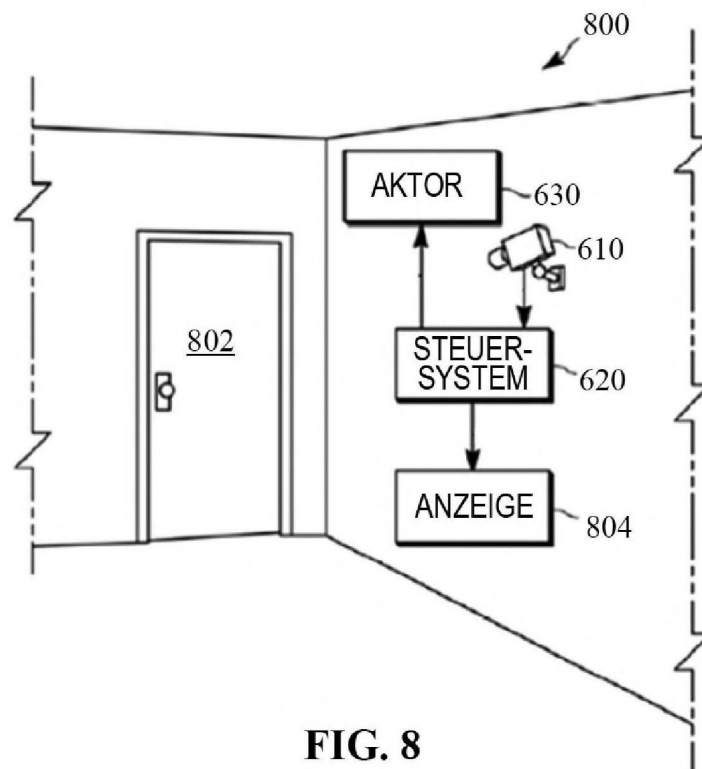


FIG. 8

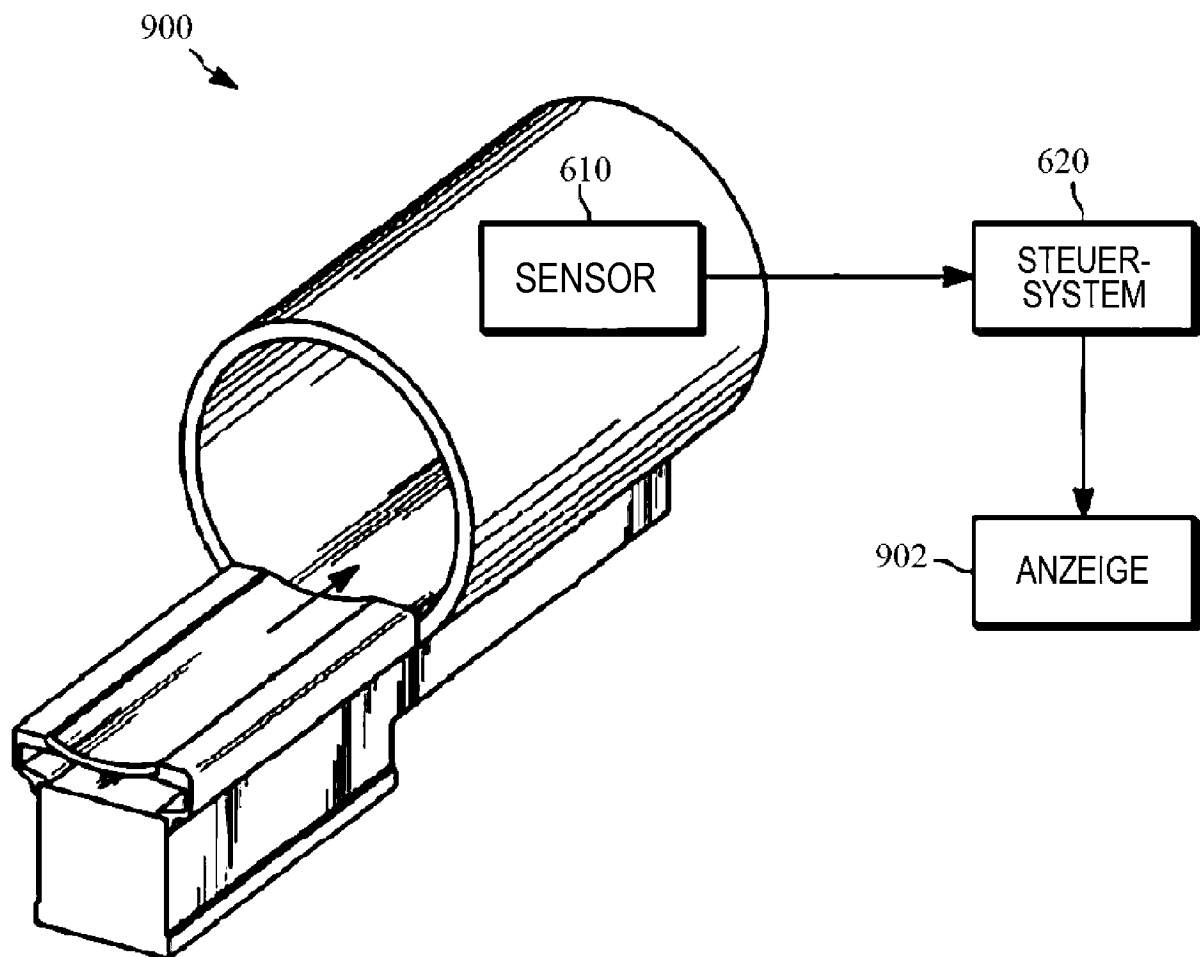


FIG. 9