US 20250265852A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2025/0265852 A1

Huang et al. (43) Pub. Date: Aug. 21, 2025

(54) **COMPUTER IMPLEMENTED METHOD, A DATASTRUCTURE AND A DEVICE FOR FINDING A MATCHED SEMANTIC NAME FOR A REGION OF A DIGITAL IMAGE OR FOR TRAINING, IN PARTICULAR A TRANSFORMER DECODER AND/OR A PIXEL DECODER, FOR FINDING A MATCHED SEMANTIC NAME FOR A REGION OF A DIGITAL IMAGE**

(71) Applicant: **Robert Bosch GmbH**, Stuttgart (DE)

(72) Inventors: **Haiwen Huang**, Tübingen (DE); **Dan Zhang**, Leonberg (DE)

(21) Appl. No.: **19/048,292**

(22) Filed: **Feb. 7, 2025**

(30) **Foreign Application Priority Data**

Feb. 15, 2024 (EP) 24 15 7956.4

Publication Classification

(51) **Int. Cl.**

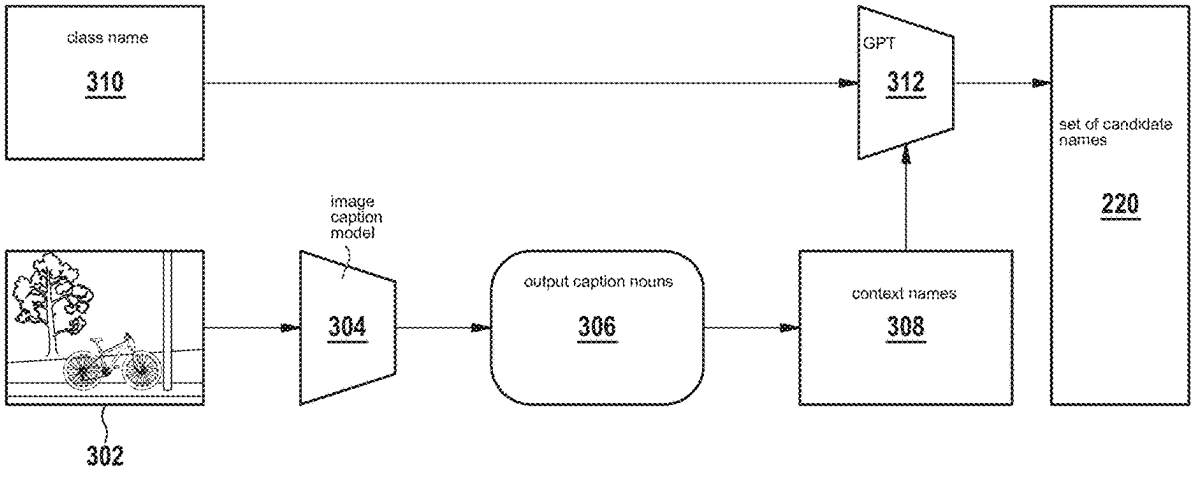
G06V 20/70 (2022.01)
G06T 9/00 (2006.01)
G06V 10/25 (2022.01)
G06V 10/44 (2022.01)
G06V 10/764 (2022.01)

(52) **U.S. Cl.**

CPC ***G06V 20/70*** (2022.01); ***G06T 9/00*** (2013.01); ***G06V 10/25*** (2022.01); ***G06V 10/44*** (2022.01); ***G06V 10/764*** (2022.01)

(57) **ABSTRACT**

A device, a datastructure, and computer implemented methods for finding a matched semantic name for a region of a digital image and for training for finding a matched semantic name for a region of a digital image. The method for training includes providing the digital image and a class name and an indicator that identifies the region in the digital image, providing a set of candidate names depending on the class name, determining an encoding of the digital image, determining multi-scale features depending on the encoding of the digital image, determining an embedding of the candidate name, determining an output including a predicted indicator and class for the respective candidate name depending on an output embedding of a transformer decoder for the embedding of the candidate name, the multi-scale features, and the indicator, determining per-pixel features, and training the transformer decoder depending on a loss.



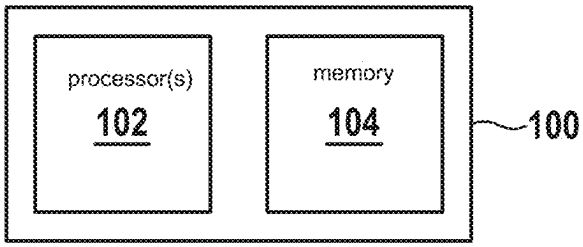


Fig. 1

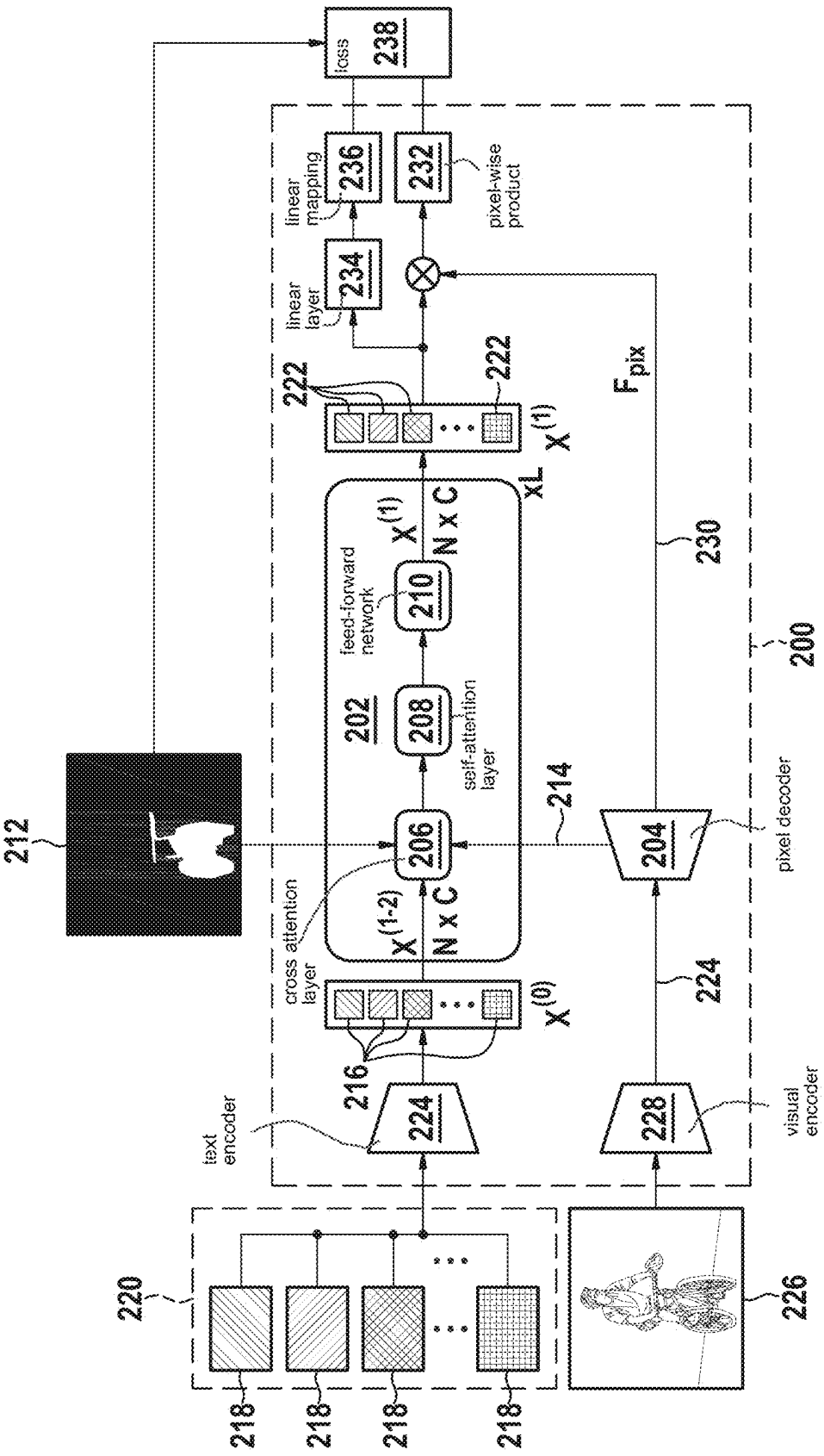

loss
238
linear mapping
236
pixel-wise product
232
linear layer
234
222
222
$X^{(1)}$
xL
feed-forward network
210
N x C
$X^{(1)}$
202
208
self-attention layer
214
212
206
cross attention layer
$X^{(1-2)}$
N x C
$X^{(0)}$
216
224
text encoder
220
218
218
218
218
F_{pix}
230
200
204
pixel decoder
224
228
visual encoder
226

Fig. 2

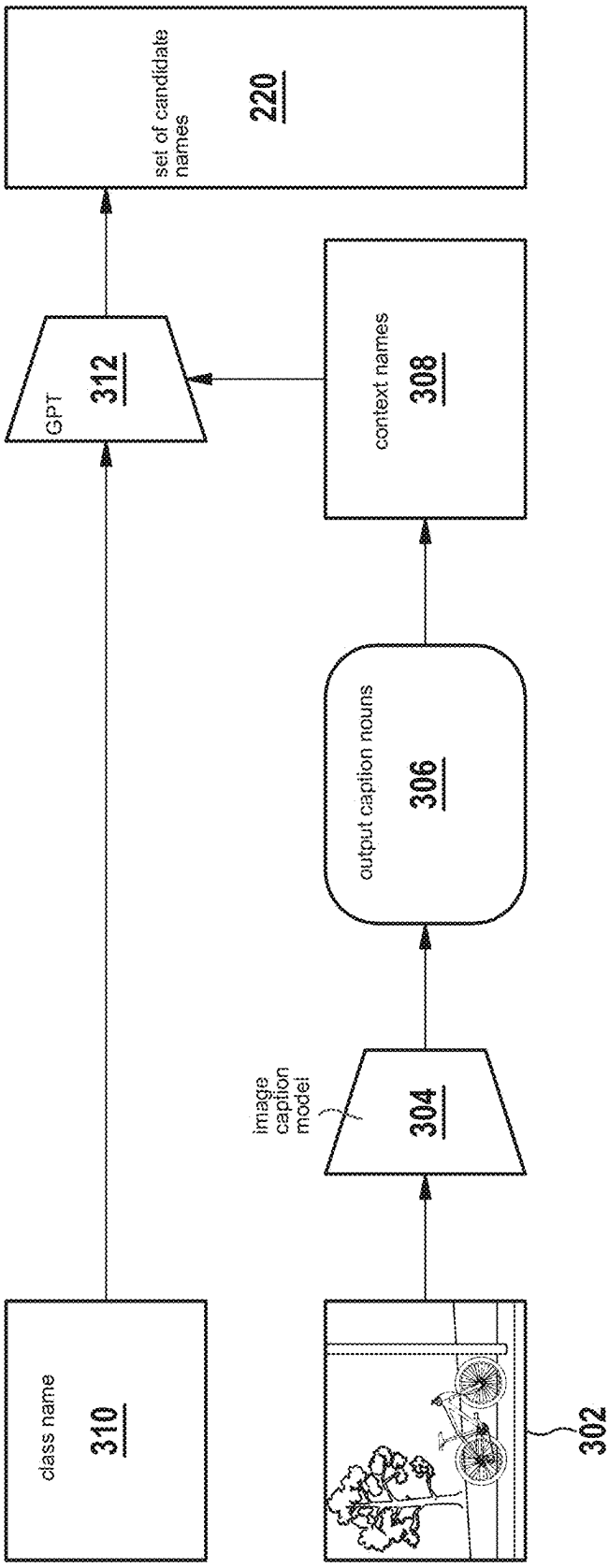

set of candidate names
220
GPT
312
context names
308
output caption nouns
306
image caption model
304
class name
310
302

Fig. 3

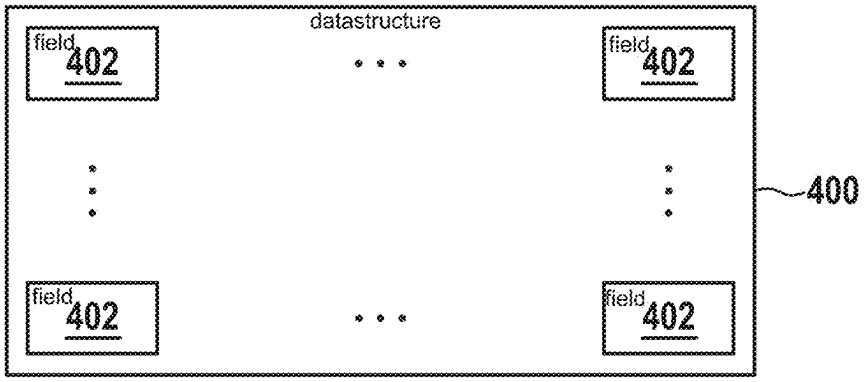


Fig. 4

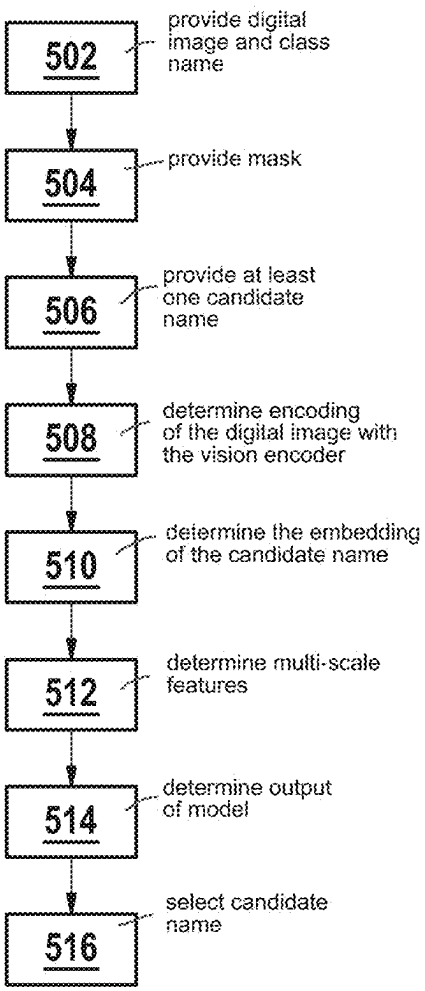


Fig. 5

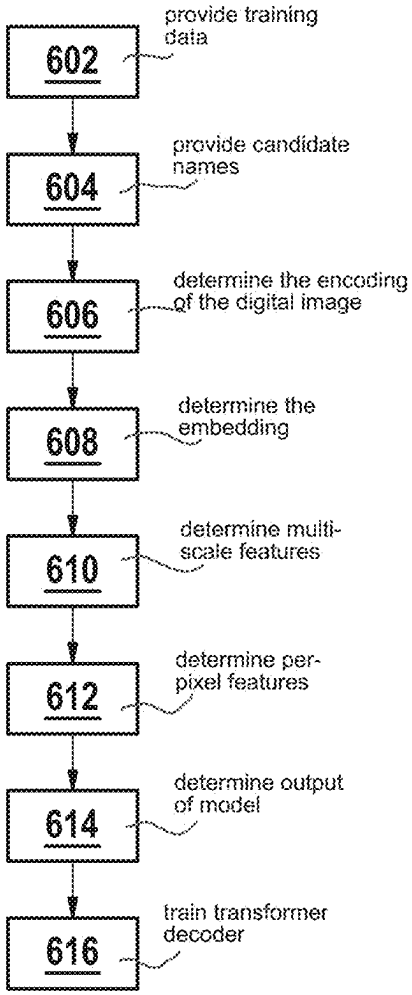


Fig. 6

COMPUTER IMPLEMENTED METHOD, A DATASTRUCTURE AND A DEVICE FOR FINDING A MATCHED SEMANTIC NAME FOR A REGION OF A DIGITAL IMAGE OR FOR TRAINING, IN PARTICULAR A TRANSFORMER DECODER AND/OR A PIXEL DECODER, FOR FINDING A MATCHED SEMANTIC NAME FOR A REGION OF A DIGITAL IMAGE

CROSS REFERENCE

[0001] The present application claims the benefit under 35 U.S.C. § 119 of European Patent Application No. EP 24 15 7956.4 filed on Feb. 15, 2024, which is expressly incorporated herein by reference in its entirety.

BACKGROUND INFORMATION

[0002] The present invention relates to a computer implemented method, a datastructure, and a device for finding a matched semantic name for a region of a digital image or for training, in particular a transformer decoder and/or a pixel decoder, for finding a matched semantic name for a region of a digital image.

[0003] Transformer decoder with masked attention assign different semantic names to different regions of a digital image.

SUMMARY

[0004] The computer implemented methods, datastructure and device according to the the present invention provide a refinement of the semantic name that is given to a region of a digital image.

[0005] According to an example embodiment of the present invention, a computer implemented method for finding a matched semantic name for a region of a digital image comprises providing the digital image and a class name, providing an indicator, in particular a bounding box or a mask, that identifies the region in the digital image, providing a set of candidate names depending on the class name, determining an encoding of the digital image, determining multi-scale features depending on the encoding of the digital image, determining an embedding of the candidate name, determining an output comprising a predicted indicator, in particular a predicted bounding box or a predicted mask, and a predicted class for the respective candidate name depending on an output embedding of a transformer decoder for the embedding of the candidate name, the multi-scale features and the indicator, and selecting the candidate name from the set that is associated with the predicted indicator that reproduces the indicator better than the other predicted indicators as semantic name for the region of the digital image depending on the output. This method refines the names per region and is thus usable for digital images with complex scenes.

[0006] According to an example embodiment of the present invention, providing the candidate name may comprise querying a promptable language model, in particular a Generative Pretrained Transformer (GPT), to output the candidate names for the class name. The GPT can generate multiple reasonable names.

[0007] The method of the present invention may comprise providing the digital image with a caption or determining a caption depending on the digital image, in particular with an image caption model, and wherein providing the candidate name comprises selecting the candidate names for the class name from the caption. The caption comprises multiple reasonable names.

[0008] According to an example embodiment of the present invention, the method may comprise determining the encoding of the digital image with a vision encoder, in particular the Contrastive Language-Image Pre-training (CLIP) image encoder or the method DINO, and/or determining the embedding of the candidate name with a text encoder, in particular the Contrastive Language-Image Pre-training text encoder. A paired vision encoder and text encoder, e.g. CLIP may be used. It is not required to use a paired vision encoder and text encoder. For example, DINO may be used as vision encoder and the text encoder part of CLIP may be used.

[0009] According to an example embodiment of the present invention, the method may comprise determining the output embedding with the transformer decoder, wherein the transformer decoder comprises a masked cross attention layer with an input for the embedding of the candidate name and the multi-scale features, and the mask, wherein the masked cross attention layer is followed by a self-attention layer, wherein the self-attention layer is followed by a feed-forward network, wherein the feed-forward network is configured to output the output embedding. This order reduces the computational resources required for implementing the transformer decoder.

[0010] According to an example embodiment of the present invention, the method may comprise determining per-pixel features depending on the multi-scale features, in particular determining the multi-scale features and the per-pixel features with a pixel decoder depending on the encoding of the digital image.

[0011] According to an example embodiment of the present invention, a computer implemented method for training for finding a matched semantic name for a region of a digital image comprises providing the digital image and a class name and a mask that identifies the region in the digital image, providing a candidate name depending on the class name, determining an encoding of the digital image, determining multi-scale features depending on the encoding of the digital image, determining an embedding of the candidate name, determining an output comprising a predicted indicator, in particular a predicted bounding box or a predicted mask, and a predicted class for the respective candidate name depending on an output embedding of a transformer decoder for the embedding of the candidate name, the multi-scale features and the indicator, determining per-pixel features depending on the multi-scale features, and training the transformer decoder depending on a loss that comprises a difference between the indicator and the predicted indicator that reproduces the indicator better than the other predicted indicators. This method trains the transformer decoder to refine the names per region and is thus usable for digital images with complex scenes.

[0012] According to an example embodiment of the present invention, the method for training may comprise determining the output embedding for a plurality of candidate names for the same digital image and mask, wherein the loss comprises a, in particular a normalized, linear mapping of the output embeddings.

[0013] According to an example embodiment of the present invention, the method for training may comprise determining the multi-scale features and the per-pixel features

with a pixel decoder depending on the encoding of the digital image, and training the pixel decoder depending on the loss.

[0014] According to an example embodiment of the present invention, the method for training may comprise determining the encoding of the digital image with a visual encoder, determining the embedding of the candidate name with a text encoder, and maintaining the visual encoder and/or the text encoder unchanged in the training. This means pre-trained, off the shelf text encoder or visual encoder may be used.

[0015] According to an example embodiment of the present invention, a device for finding a matched semantic name for a region of a digital image or for training, in particular a transformer decoder and/or a pixel decoder, for finding a matched semantic name for a region of a digital image, comprises at least one processor and at least one memory, wherein the at least one memory stores instructions that, when executed by the at least one processor, cause the device to execute the method for finding the matched semantic or for training.

[0016] According to an example embodiment of the present invention, a datastructure for finding a matched semantic name for a region of a digital image or for training, in particular a transformer decoder, for finding a matched semantic name for a region of a digital image, comprises at least one data field for the digital image, for a class name, for a mask that identifies the region in the digital image, for a candidate name that is determined depending on the class name, for an encoding of the digital image, for multi-scale features that are determined depending on the encoding of the digital image, for an embedding of the candidate name, for an output embedding, in particular of the transformer decoder for the embedding of the candidate name, the multi-scale features and the indicator, and for an output representing the candidate name that is determined depending on the output embeddings.

[0017] In particular for training, according to an example embodiment of the present invention, the datastructure may comprise at least one data field for per-pixel features that are determined depending on the multi-scale features, and for a loss that comprises a pixel-wise multiplication of the per-pixel features with the output representing the candidate name.

[0018] According to an example embodiment of the present invention, the datastructure may comprise at least one data field for a, in particular normalized, linear mapping of the output embeddings.

[0019] Further advantageous embodiments of the present invention may be derived from the following description and the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] FIG. **1** schematically depicts a device for finding a matched semantic name for a region of a digital image or for training, in particular a transformer decoder and/or a pixel decoder for finding a matched semantic name for a region of a digital image, according to an example embodiment of the present invention.

[0021] FIG. **2** schematically depicts an overview of a training of the transformer decoder and/or the pixel decoder, according to an example embodiment of the present invention.

[0022] FIG. **3** schematically depicts an overview of a candidate name generation, according to an example embodiment of the present invention.

[0023] FIG. **4** schematically depicts a datastructure, according to an example embodiment of the present invention.

[0024] FIG. **5** depicts a flowchart with steps of a method for finding a matched semantic name for a region of a digital image, according to an example embodiment of the present invention.

[0025] FIG. **6** depicts a flow chart with steps of a method for the training, according to an example embodiment of the present invention.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENTS

[0026] FIG. **1** schematically depicts a device **100**. The device **100** comprises at least one processor **102** and at least one memory **104**.

[0027] According to an example, the device **100** is configured for finding a matched semantic name for a region of a digital image.

[0028] According to an example, the device is configured for training, in particular a transformer decoder and/or a pixel decoder, for finding a matched semantic name for a region of a digital image.

[0029] The at least one memory **104** stores instructions that, when executed by the at least one processor **102**, cause the device to execute a method for finding the matched semantic name or for training.

[0030] FIG. **2** schematically depicts a model **200** for finding the matched semantic name. The model **200** comprises a transformer decoder **202** and a pixel decoder **204**.

[0031] FIG. **2** schematically depicts an overview of the training of the transformer decoder **202** and/or the pixel decoder **204**.

[0032] The transformer decoder **202** in the example is an artificial neural network. The transformer decoder **202** in the example comprises L layers. The transformer decoder **202** in the example comprises in each layer **1** a cross attention layer **206**, a self-attention layer **208**, and a feed forward network **210**. The cross attention layer **206** is followed by the self-attention layer **208**. The self-attention layer **208** is followed by the feed forward network **210**. In the example, followed means that the output of the cross attention layer **206** is the input of the self-attention layer **208**. In the example, the output of the self-attention layer **208** is the input of the feed forward layer **210**. There may be layers in between these layers.

[0033] The cross attention layer **206** comprises an input for a mask **212** and multi-scale features **214** from the pixel decoder **204** and an embedding **216** of a candidate name **218**.

[0034] The mask **212** is an example for an indicator indicating the region. Instead of the mask **212**, the indicator may be a bounding box.

[0035] In the example, a set **220** of candidate names **218** is provided. In the example, the set **220** of candidate names **218** is associated with a class c_i. The candidate name **218** is provided from the set **220** of candidate class names **218**.

[0036] The transformer decoder **202** is configured to output, for an embedding **216** of a candidate class name **218** an output embedding **222**. The model **200** is configured to determine an output of the model depending on the output

embedding **222**. The output of the model comprises a predicted mask, and a predicted class for the respective candidate name **218**.

[0037] In case that the indicator is the bounding box, the model **200** may be configured to output the output of the model comprising a predicted bounding box instead of the predicted mask.

[0038] The transformer decoder **202** is configured to determine the output **222** iteratively in L iterations. According to the example, the embedding **216** of the candidate name **218** is an initial input $X^{(0)}$ and the output **222** is the result of a last iteration $X^{(L)}$. In the example, the input $x^{(I-1)}$ of an iteration and the output $X^{(I)}$ of an iteration has the dimensions N×C.

[0039] The transformer decoder **202** in the example is configured as described in Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), **2022** (Mask2Former).

[0040] The embedding **216** of the candidate class name **218** in the example is determined with a text encoder **224**. The text encoder **224** is configured to encode the candidate name **218** to its embedding **216**. The text encoder **224** may be CLIP. The text encoder **224** in the example is pre-trained.

[0041] FIG. **2** depicts a plurality of candidate class names **218**, their respective embeddings **216** and the outputs **222** of the transformer decoder **202** for the embedding **216** of the respective candidate name **218**.

[0042] The pixel decoder **204** is configured to determine the multi scale features **214** from an embedding **224** of a digital image **226**. In the example, the embedding **224** of the digital image **226** is determined with a visual encoder **228**.

[0043] The visual encoder **228** may be CLIP or DINO. The visual encoder **228** in the example is pre-trained.

[0044] The pixel decoder **204** is configured to output per-pixel features **230** of dimension H×W×C, denoted as F_{pix}.

[0045] For the output **222** of the transformer decoder **202** of an iteration 1 an intermediate mask $\hat{M}^{(I)}$ is determined as a pixel-wise product **232** of the per-pixel features **230** and the respective candidate name **218**.

[0046] The pixel-wise product **232** in the example is an intermediate mask $\hat{M}^{(I)}$ determined as:

$$\hat{M}^{(I)} = \text{sigmoid}\left(X^{(I)} * F_{pix}\right)$$

[0047] wherein * refers to the pixel wise multiplication.

[0048] For the output **222** of the transformer decoder **202** of an iteration, a predicted class $\hat{c}^{(I)}$ is determined with a linear layer **234** that outputs a linear mapping **236**, i.e. the class $\hat{c}^{(I)}$:

$$\hat{c}^{(I)} = \text{softmax}\left[\text{Linear}\left(X^{(I)}\right)\right]$$

[0049] The final prediction, i.e., the final mask $\hat{M}^{(L)}$ and the final class $\hat{c}^{(L)}$ in the output of the model **200**, is made depending on the result $X^{(L)}$ of the last iteration L.

[0050] The mask **212** of a training data point i is denoted M_i. The mask M_i identifies a region in the digital image **226**.

[0051] For training, a loss **238**, denoted as L_i for the training data point i, is used that depends on the mask M_i, the best predicted class $\hat{c}_{best}$ of the predicted classes $\hat{c}^{(I)}$ and the best predicted mask $\hat{M}_{best}$ of the masks $\hat{M}^{(I)}$:

$$L_i = L_{mask}\left(M_i, \hat{M}_{best}\right) + L_{class}(c_i, \hat{c}_{best})$$

[0052] wherein L_{mask} is the mask localization function and L_{class} is the classification loss function in Mask2Former.

[0053] With training data comprising pairs of ground-truth mask M_i and class C_i, and with both the vision and text encoder kept frozen, the transformer decoder **202** is trained together with the pixel decoder **204** for segmentation. To recover the ground-truth mask M_i and class c_i the transformer decoder **202** makes multiple mask and class predictions with candidate names from the class c_i. In the example, the prediction with the highest Intersection over Union (IoU) with M_i is selected for loss computation.

[0054] According to an example, to provide extra supervision on the name quality, the candidate names in the training data may be appended with a "negative" name randomly selected from a different category than the ground-truth class c_i. If a "negative" name scores the highest IoU, the predictions are supervised with an empty mask and a "void" class. The void class is one extra class in addition to the training classes:

$$L_i = L_{mask}\left(0, \hat{M}_{best}\right) + L_{class}(K+1, \hat{c}_{best})$$

[0055] Having both the positive and negative supervision, we effectively incentivize the model to favor names of high quality and penalize those of low quality, thereby aiding in the accurate identification of the best-matching names for each segment.

[0056] According to an example, after the training with the training data including the negative names, the model **200** is run on the training data with ground-truth classes c_i and without the negative names again, i.e. using ground-truth masks M_i as attention biases and leaving out the negative name appending. In the example, the candidate name that generates the mask prediction with the highest IoU with the ground-truth mask M_i is associated to the ground-truth mask M_i.

[0057] According to an example, for the ground-truth class c, the candidate names that the model **200** selects for the training data are aggregated and sorted based, e.g., on their frequency in matching with ground-truth masks M_i. To retain the most precise names and discard less common, potentially noisy candidate names, the top-ranking candidate names may be selected, such that they can cover e.g. at least 90% total frequencies. The top-ranking candidate names are the renovated names for that ground-truth class c_i.

[0058] The top-ranking candidate names for the ground-truth classes c_i are more descriptive and precise than the original class names of the ground-truth classes c_i. When using the top-ranking candidate names as text prompts, a pre-trained open-vocabulary model achieves much higher segmentation performance than using the original class names.

[0059] FIG. **3** schematically depicts an overview of a candidate name generation.

[0060] The candidate name generation is based on digital images of the classes c_i.

[0061] A digital image **302** for a class c_i is provided to an image caption model **304**. The caption model **304** is configured to output caption nouns **306** for the digital image **302** from the set. According to an example, the image caption model **304** outputs the nouns tree, pavement, hedge, mountain bike, bicycle rack, grass, sidewalk, building, saddlebag. In the example, the caption nouns **306** are determined for a set of digital images of the class c_i comprising the digital image **302**.

[0062] The nouns from the caption nouns that the image caption model **304** generates for the set of digital images of the class c_i are aggregated and sorted to context names **308** for the class c_i. According to an example, the context names **308** comprise the nouns building, road, person, park, pavement, car, walk, city street, bicycle, street corner.

[0063] The class c_i is associated with a class name **310**. In the example, the set **220** of candidate names **218** is provided for the class c_i that is associated with the class name **310**. According to the example, the class name bicycle is associated with the class c_i.

[0064] The class name **310** of the class c_i and the context names **308** generated for the class c_i are input to a Generative Pretrained Transformer, GPT, **312**, that is configured to output the set **220** of candidate names **218** for the class c_i.

[0065] In the example, the GPT **312** outputs city bike, bicycle, road bicycle, tandem bike, cruiser, pavement bike, recreational bike, mountain bike, pedestrian bicycle.

[0066] According to an example of K classes C_i, the candidate names **218** are determined for the respective class c_i from a set of digital images and the class name for the respective class c_i.

[0067] This means, the GPT **312** selects the candidate names from the caption of the digital images. The caption comprises multiple reasonable names. The candidate names may be selected from the caption directly as well.

[0068] FIG. **4** schematically depicts a datastructure **400** for finding the matched semantic name for a region of the digital image **226**.

[0069] The datastructure **400** comprises at least one data field **402** for the digital image **226**, for the class name **310**, for the mask **212** that identifies the region in the digital image **226**, for at least one candidate name **218** that is determined depending on the class name **310**, for the encoding **214** of the digital image **226**, for multi-scale features **214** that are determined depending on the encoding **214** of the digital image **226**, for an embedding **216** of the candidate name **218**, and for an output **222** representing the candidate name **218** that is determined depending on the embedding **216** of the candidate name **218**, the multi-scale features **214** and the mask **212**.

[0070] The datastructure **400** may in particular for training comprises at least one data field **402** for per-pixel features **230** that are determined depending on the multi-scale features **214**, and for a loss **238** that comprises a pixel-wise multiplication **232** of the per-pixel features **230** with the output **222** representing the candidate name **218**.

[0071] The datastructure **400** may comprise at least one data field **402** for a, in particular normalized, linear mapping **236** of the outputs **222**.

[0072] FIG. **5** depicts a flow chart comprising steps of a computer implemented method for finding a matched semantic name for a region of the digital image **226**.

[0073] The steps of the method are described for the digital image **226** that is associated to the class name **310**.

[0074] The method comprises a step **502**.

[0075] The step **502** comprises providing **502** the digital image **226** and the class name **310**.

[0076] In the example, the class name **310** is associated to the digital image **226**. For example, a classification of the digital image **226** is executed that associates the class name **310** to the digital image **226**.

[0077] The method comprises a step **504**.

[0078] The step **504** comprises providing the mask **212**. The mask **212** identifies the region in the digital image **226**.

[0079] The method comprises a step **506**.

[0080] The step **506** comprises providing at least one candidate name **218** depending on the class name **310**. In the example the set **220** of candidate names **218** is provided.

[0081] According to an example, providing the at least one candidate name **218** comprises querying the Generative Pretrained Transformer **312** to output the set **220** of candidate names **218** for the class name **310**. Providing the at least one candidate name may comprise selecting the candidate names **218** for the class name **310** from the caption **306**.

[0082] According to an example, the method comprises providing the digital image **226** with the caption. The caption for example comprises the caption nouns **306**. According to an example, the method comprises determining the caption, e.g., the caption nouns **306**, depending on the digital image **226**.

[0083] The method may comprise determining the caption with a caption model. In an example, the caption nouns **306** are determined with the image caption model **304**.

[0084] The method comprises a step **508**.

[0085] The step **508** comprises determining the encoding **214** of the digital image **226**.

[0086] The method may comprise determining the encoding **224** of the digital image **226** with the vision encoder **228**. The vision encoder **228** may be the CLIP neural network or the method DINO.

[0087] The method comprises a step **510**.

[0088] The step **510** comprises determining the embedding **216** of the candidate name **218**. The method may comprise or determining the embedding **216** of the candidate name **218** with the text encoder **224**, in particular the CLIP neural network.

[0089] The step **510** is executed for the candidate names **218** in the set **220** of candidate names **218**.

[0090] The method comprises a step **512**.

[0091] The step **512** comprises determining multi-scale features **214** depending on the encoding **214** of the digital image **226**.

[0092] The step **512** comprises determining per-pixel features **230** depending on the multi-scale features **214** depending on the encoding **224** of the digital image **226**.

[0093] According to an example, the multi-scale features **214** and the per-pixel features **230** are determined with the pixel decoder **204**.

[0094] In the example, the per-pixel features **230** are determined for a plurality of candidate names **218**.

[0095] The method comprises a step **514**.

[0096] The step **514** comprises determining the output of the model **200**. Determining the output of the model **200**

comprises determining the output **222** for the respective candidate names **218** of the set **220** depending on the respective embeddings **216** of the candidate names **218** of the set **220**, the multi-scale features **214** and the mask **212**.

[0097] Determining the output of the model **200** comprises determining the output **222** of the transformer decoder **202** and the output of the model **200** in iterations.

[0098] The output of the model **200** is the final prediction, i.e., the final mask $\hat{M}^{(L)}$ and the final class $\hat{c}^{(L)}$ that is made depending on the result $X^{(L)}$ of the last iteration L.

[0099] The output **222** in the example is determined with the transformer decoder **202**.

[0100] This means, for at least one candidate name **218**, the output **222** is determined with the transformer decoder **202** comprising the masked cross attention layer **206** with the input for the embedding **216** of the candidate name **218** and the multi-scale features **214**, and the mask **212**.

[0101] This means, the output **222** is determined for the candidate names **218** of the set **220** of candidate names **218** for the digital image **226** and the mask **212**.

[0102] The method comprises a step **516**.

[0103] The step **516** comprises selecting the candidate name **218** that is associated with the predicted mask M_{best} that reproduces the mask **212** better than the other predicted masks $\hat{M}_{other}=\hat{M}^{(L)}\setminus\hat{M}_{best}$ as semantic name for the region of the digital image **226** depending on the output of the model **200**.

[0104] This means, in the set **220** of candidate names, the candidate name **218** that is associated with the predicted indicator that reproduces the indicator **212** better than the other predicted indicators is selected as semantic name for the region of the digital image **226** depending on the output of the model **200**.

[0105] According to an example, the intersection over union of respective predicted mask $\hat{M}^{(l)}$ and the mask **212** is computed as measure for the correlation. The measure for correlation, e.g., the intersection over union, indicates how well a predicted mask $\hat{M}^{(l)}$ reproduces the mask **212**

[0106] The measure for the correlation may be determined for the candidate names **218** in the set **220** of candidate names **218**.

[0107] According to an example, the at least one candidate name **218** is selected as semantic name for the region of the digital image **226** that results in the measure that indicates a higher correlation, e.g., a higher intersection over union, than the other measures.

[0108] According to an example, the candidate name **218** is selected that results in the measure that indicates the largest correlation.

[0109] FIG. **6** depicts a flow chart with steps of a method for the training.

[0110] The method for training comprises a step **602**.

[0111] The step **602** comprises providing training data.

[0112] The training data comprises a plurality of digital images associated to a respective ground truth class and a respective ground truth mask.

[0113] The training data comprises a plurality of ground-truth classes C_i and ground-truth masks M_i.

[0114] The class name **310** is an example for the ground-truth classes C_i. The mask **212** is an example for the ground-truth masks M_i.

[0115] The method for training comprises running the model **200** as described for the exemplary class name **310** and the exemplary ground truth mask **212**.

[0116] The training data for example comprises the digital image **226** and the class name **310** and the mask **212** that identifies the region in the digital image **226**.

[0117] The method for training comprises executing the following steps for each of the digital images of the training data. The following steps are described by way of example of the digital image **226** that is associated with the class name **310**.

[0118] The method for training comprises a step **604**.

[0119] The step **604** comprises providing candidate names **218** depending on the class name **310**.

[0120] The step **604** comprises providing the set **220** of candidate names **218** depending on the class name **310**.

[0121] The candidate class names **218** may be determined as described in step **506**.

[0122] According to an example, for the ground-truth class **310**, the candidate names **218** that the model **200** selects for the training data from the set **220** of candidate names **218** are aggregated and sorted based, e.g., on their frequency in matching with ground-truth masks **212**. To retain the most precise names and discard less common, potentially noisy candidate names, the top-ranking candidate names may be selected, such that they can cover e.g. at least 90% total frequencies. The top-ranking candidate names are the renovated names for that ground-truth class **310**.

[0123] The method for training comprises a step **606**.

[0124] The step **606** comprises determining the encoding **224** of the digital image **226**.

[0125] The method for training may comprise determining the encoding **224** of the digital image **226** with the visual encoder **228**.

[0126] The method for training comprises a step **608**.

[0127] The step **608** comprises determining the embedding s **216** of the candidate names **218** in the set **220** of candidate names **218**.

[0128] The method for training may comprise determining the embedding **216** of the candidate name **218** with the text encoder **224**.

[0129] The method for training comprises a step **610**.

[0130] The step **610** comprises determining multi-scale features **214** depending on the encoding **214** of the digital image **226**.

[0131] The method for training comprises a step **612**.

[0132] The step **612** comprises determining the per-pixel features **230** depending on the multi-scale features **214**.

[0133] According to an example, the multi-scale features **214** and the per-pixel features **230** are determined with the pixel decoder **204** depending on the encoding **214** of the digital image **226**.

[0134] The method for training comprises a step **614**.

[0135] The step **614** comprises determining the output of the model **200**. Determining the output of the model **200** comprises determining the output **222** representing the candidate names **218**.

[0136] Determining the output of the model **200** comprises determining the output **222** of the transformer decoder **202** and the output of the model **200** in iterations.

[0137] The output of the model **200** is the final prediction, i.e., the final mask $\hat{M}^{(L)}$ and the final class $\hat{c}^{(L)}$ that is made depending on the result $X^{(L)}$ of the last iteration L.

[0138] The output **222** for a respective candidate name **218** is determined in the example with the transformer

decoder **202** depending on the embedding **216** of the respective candidate name **218**, the multi-scale features **214** and the mask **212**.

[0139] This means, the output **222** may be determined for a plurality of candidate names **218** for the same digital image **226** and mask **212**.

[0140] The method for training comprises a step **616**.

[0141] The step **616** comprises training the transformer decoder **202** depending on the loss **238**. The step **616** may comprise training the pixel decoder **204** depending on the loss **238**.

[0142] In the example, the candidate name that generates the mask prediction with the highest IoU with the ground-truth mask M_i is used to determine the loss **238**.

[0143] The method for training may comprise maintaining the visual encoder **228** and/or the text encoder **224** unchanged in the training.

[0144] The class name **310** is an example for the ground-truth classes C. The mask **212** is an example for the ground-truth masks M_i.

[0145] The method for training comprises running the model **200** as described in the following steps of the method for training for a plurality of ground-truth classes c_i and ground-truth masks M_i.

[0146] According to an example, the method for training comprises running the model **200** with the training data including the negative names.

[0147] According to an example, after the training with the training data including the negative names, the model **200** is run on the training data with ground-truth classes c_i and without the negative names again, i.e. using ground-truth masks M_i as attention biases and leaving out the negative name appending.

[0148] The method for training may comprise determining the parameters of the transformer decoder **202** and/or the pixel decoder **204** with a gradient descent method based on the loss that is determined for the training data.

What is claimed is:

1. A computer implemented method for finding a matched semantic name for a region of a digital image, the method comprising the following steps:

providing the digital image and a class name;

providing an indicator, including a bounding box or a mask, the indicator identifying the region in the digital image;

providing a set of candidate names depending on the class name;

determining an encoding of the digital image;

determining multi-scale features depending on the encoding of the digital image;

determining an embedding of the candidate name;

determining an output including a predicted indicator, including a predicted bounding box or a predicted mask, and a predicted class for the respective candidate name, depending on an output embedding of a transformer decoder for the embedding of the candidate class name, the multi-scale features, and the indicator; and

selecting a candidate name from the set that is associated with the predicted indicator that reproduces the indicator better than the other predicted indicators, as the semantic name for the region of the digital image depending on the output.

2. The method according to claim **1**, wherein the providing of the set of candidate names includes querying a promptable language model, including a Generative Pre-trained Transformer, to output the candidate names for the class name.

3. The method according to claim **1**, further comprising providing the digital image with a caption or determining a caption, depending on the digital image, using an image caption model, wherein the providing of the set of candidate names includes selecting the candidate names for the class name from the caption.

4. The method according to claim **1**, further comprising (i) determining the encoding of the digital image using a vision encoder including a Contrastive Language-Image Pre-training (CLIP) neural network or a DINO method, and/or (ii) determining the embedding of the candidate name with a text encoder, including a Contrastive Language-Image Pre-training (CLIP) neural network.

5. The method according to claim **1**, further comprising determining the output embedding with the transformer decoder, wherein the transformer decoder includes a masked cross attention layer with an input for the embedding of the candidate name and the multi-scale features, and the mask, wherein the masked cross attention layer is followed by a self-attention layer, wherein the self-attention layer is followed by a feed-forward network, wherein the feed-forward network is configured to output the output embedding.

6. The method according to claim **1**, further comprising determining per-pixel features depending on the multi-scale features, including determining the multi-scale features and the per-pixel features using a pixel decoder depending on the encoding of the digital image.

7. A computer implemented method for training for finding a matched semantic name for a region of a digital image, comprising the following steps:

providing the digital image and a class name and an indicator, the indicator including a bounding box or a mask, the indicator identifying the region in the digital image;

providing a set of candidate names depending on the class name;

determining an encoding of the digital image;

determining multi-scale features depending on the encoding of the digital image;

determining an embedding of the candidate name;

determining an output including a predicted indicator, including a predicted bounding box or a predicted mask, and a predicted class for the respective candidate name depending on an output embedding of a transformer decoder for the embedding of the candidate name, the multi-scale features, and the indicator;

determining per-pixel features depending on the multi-scale features; and

training the transformer decoder depending on a loss that includes a difference between the indicator and the predicted indicator that reproduces the indicator better than other predicted indicators.

8. The computer implemented method according to claim **7**, further comprising determining an output embedding for a plurality of candidate names for the same digital image and mask, wherein the loss includes a normalized, linear mapping of the output embedding for the plurality of candidate names.

9. The computer implemented method according to claim 7, further comprising:

determining the multi-scale features and the per-pixel features with a pixel decoder depending on the encoding of the digital image; and

training the pixel decoder depending on the loss.

10. The computer implemented method according to claim 7, further comprising:

determining the encoding of the digital image using a visual encoder;

determining the embedding of the candidate name using a text encoder; and

maintaining the visual encoder and/or the text encoder unchanged in the training.

11. A device for finding a matched semantic name for a region of a digital image or for training a transformer decoder and/or a pixel decoder, for finding a matched semantic name for a region of a digital image, the device comprising:

at least one processor; and

at least one memory, wherein the at least one memory stores instructions that, when executed by the at least one processor, cause the device to perform:

providing the digital image and a class name and an indicator, the indicator including a bounding box or a mask, the indicator identifying the region in the digital image,

providing a set of candidate names depending on the class name,

determining an encoding of the digital image,

determining multi-scale features depending on the encoding of the digital image,

determining an embedding of the candidate name,

determining an output including a predicted indicator, including a predicted bounding box or a predicted mask, and a predicted class for the respective candidate name depending on an output embedding of a transformer decoder for the embedding of the candidate name, the multi-scale features, and the indicator,

determining per-pixel features depending on the multi-scale features, and

training the transformer decoder depending on a loss that includes a difference between the indicator and the predicted indicator that reproduces the indicator better than other predicted indicators.

12. A datastructure for finding a matched semantic name for a region of a digital image or for training a transformer decoder for finding a matched semantic name for a region of a digital image, the data structure comprising:

at least one data field for the digital image, for a class name, for an indicator, including a bounding box or a mask, that identifies the region in the digital image, for a candidate name that is determined depending on the class name, for an encoding of the digital image, for multi-scale features that are determined depending on the encoding of the digital image, for an embedding of the candidate name, for an output embedding, including the transformer decoder for the embedding of the candidate name, the multi-scale features and the indicator, and for an output including a predicted indicator, including a predicted bounding box or a predicted mask, and a predicted class for the candidate name that is determined depending on the output embedding.

13. The datastructure according to claim 12, wherein the datastructure is for training, and the datastructure comprises at least one data field for per-pixel features that are determined depending on the multi-scale features, and for a loss that includes a difference between the indicator and the predicted indicator that reproduces the indicator better than other predicted indicators.

14. The datastructure according to claim 13, wherein the datastructure further comprises at least one data field for a r normalized, linear mapping of the output embedding.

* * * * *