Contents

1 Abstract

Keywords: Optical character recognition, Few-shot transfer learning, Vision transformers, Paleontological databases

2 Introduction

broad research area: paleontology: data analysis on fossil finds dig fossil from the ground. identify:which bone,species,time. write this on a field slip, a slice of baking sheet like paper analysis: take set of fossils, use methods for deducing eg climate, habitat, vegetation why this is interesting reactions of ecosystems to climate change what ancient worlds were like how ancient humans lived mass extinction events

now we have a stack of baking sheets in a room in kenya. paleontologists form all over the world want to solve climate change, among other problems big data methods would explode what paleoecology can do so we need to put the baking sheets on the computer to do analysis on big data what is the topic of my work? the baking sheets contain weird characters that a normal reader cannot read. my topic is to read them

given scanned images and data with bounding boxes of sentences and words where tooth denoting words are badly read, how can tooth element recognition results be improved? Goal is to have both this is what the element / nature of specimen column says (eg example here) and what/which teeth are found in this specimen in standardized format (eg example here)

new for intro

motivating why this is relevant, wsj article [17]

limitations: compute (no HPC utilized), small amount of labeled data, scope is masters thesis so very advanced techniques are not feasible.

relevance of this work: KNM is able to have way more precise dental element markings to other catalogues: previous project + this a complete solution to digitizing the handwritten data relevance of this work: any field that does: - ocr on unconventional characters - ocr where each character has a multivariate output (eg. this is an a. it is underlined could be letter and underlined /not underlined)

The direct impact of this work is an improved precision of the tooth element entries in the digitized fossil catalogues of the National Museum of Kenya, but the results are applicable to a wider domain of problems. Intuitively, the results are directly applicable to other fossil archives using similar notation: only a fine-tuning of the models to the new archive data is necessary. For other handwritten archives, the results presented can be used to improve recognition accuracy, especially in cases where the data contains characters other than latin letters or arabic numerals. Additionally, this work presents a potential solution for when the target character set can be expressed with multivariate output data. This could, for instance, be handwriting with occasional underlinings, where the bivariate output could be the letter and a boolean variable for whether the character was underlined.

The rest of this thesis is organized as follows. First, the necessary background theory is presented. For deep neural networks the following concepts are introduced: the basic network structure, how training is conducted, basic building blocks of character-recognizing network architectures, performance-improving heuristics, and transfer learning. For paleoecology, the background covers foundational ecological laws followed by a brief introduction to methods used in paleoenvironmental reconstruction, especially focusing on inferences from tooth data. As the last background section, the composition of mammal teeth is presented. Second, related work is presented, both on handwritten archive digitization and transfer learning with character-recognizer models. Next, the experimental setup is introduced, covering dataset creation, labeling and data preprocessing, followed by base model and transfer learning method selection. After this, results of the experiments are presented and discussed. Finally, the work is concluded.

3 Fundamentals on paleoecology

Nature is highly complicated -¿ models, approximate models and assumptions enable drawing conclusions from known distributions of species.

each assumption / model can be questioned but they hold as a rule. only models briefly presented here, all statements here can be questioned to some extent

idea: what fossil/dental data can be used for. how fossil/dental data is used

to highlight why accurate, fine-resolution (ie specific) and large magnitude of fossil, esp dental fossil data is genuinely useful.

chapter overview: review ecology and assumptions the analysis is based on. then, short overview of main techniques for paleoenvironmental reconstruction, the main application area of fossil data. last, mammal teeth row is presented to introduce terminology present in the data.

3.1 Basics on ecology

basic laws: theory that the data analysis relies on

Tolerances and niches (fundamental + realized): basis for environmental reconstruction [6] ch 2

tolerance = range of an environmental variable that is hospitable for the species, eg. imaginary small mammal (come up with some imaginary name) can live when avg temperature is +10-+15. Niche = set of tolerances the species has. fundamental niche = possible environments for the species, realized niche = where it lives. center of tolerances is better than the edge (ch2)

main assumption uniformitarism (the fact that tolerances constraint things has not changed) niche conservatism: Assume that nearest living relative has same tolerances now -¿ get past environment (ch3, lyman 2017),

this presents a mapping from taxa to environmental variables -¿ basis of analysis of past environments.

modern alternative to this: transfer functions: mappings from taxa data to enviroment learned using machine learning / statistical models (ch9, birks 1995)

benefit instead of tolerances/niches: they have subjective interpretation problems (book ch 9)

esp. teeth: dental ecometrics = inference of transfer functions given dental data (ch9 liu et al 2012)

next, turn to how to solve the problem of information to environment given the taxomic information to environmetal indicators mapping

3.2 Paleoenvironmental reconstruction

why: get information of what is to come with climate change (faith ch2)

definition of the problem: whan ancient habitats were like and what changes they underwent at which times (ch2)

overview main tehcniques: presence/absence, abundance, taxon free, diversity based, size clines

presence absence (ch5): dataset is list of taxa that are present. absence is also indicator but worse since might be that the species just was not preserved / found. two approaches: fix location or fix set of species. fix species: find where this set of species lives now (climate maps & areas of sympatry), this indicates that historical locality had this climate. place-fixing: analyze how which species show up in this place changes over time, reduce species showing up data to lower dimension (like pca), eg how many warm-climate vs cold-climate species show up, this gives ideas on changes in climate over time abundance get relative NISP (number of species in sample): percent of species in sample is of this taxon (grayson 1984b, ch6) do like presence absence but weigh signal according to abundance and assume abundant species lives more toward center of tolerance grayson 1981: fossil accumulation affects datasets -¿ use with caution

taxon free cornelis van der klaauw 1948 (p 160) relation of traits of animals and livelihood = ecomorphology: eg what the animal eats also eg how diverse the place is. for environment mostly diet -¿ what plants grew and habitat -¿ climate etc helps with not having to assume that closest relative tolerances were the same. (ch7) eg usage of teeth: dental microwear. calandra and merceron 2016: analyze miniscratches on the surface to get diet to get vegetation and climate oma: to conduct this you need many samples of the same bone need to know exactly which bone it is since different teeth used for different things

diversity and size clines as enviromnental indicators andrews et al 1979: how diverse also has a mapping to environment eg tropical is more diverse than arctic coarse indicators of the environment lyman 2008: strong sample size effects: bigger sample is more diverse so to do this you need equal size samples from all time periods analyzed size clines basic idea: coarse indicators for size of bone -¿ environment eg larger libs warmer climate (mayr 1970) also dental measures correlate with environment, eg mandibles (faith et al 2016). for that you need sufficient tooth samples and correct identification which tooth it is

end lesson: sample sizes and data precision are important. therefore it is superimportant to get more data / improve accuracy, which is the main goal of the digitization effort. scale sense: faith lyman ch 4 said 1000 is solid 10 000 whopping sample size dataset in question in this work has 90,000 so the value is pretty clear

3.3 Composition of mammal teeth

Fossils occur when animal / plant remains are deposited in a sediment in a way that preserves some part of its original form. Since teeth are the hardest material in animals, large fraction of found parts are teeth. Fossil finding is followed by identification to most specific taxon possible largely a technical skill (ch5), teeth are identified down to type and number, how manyeth the teeth are, counting from center to edge or other way round?? specimen can be either one tooth or fragments of the jaw bone where there are multiple teeth (markings like M1-3)

from [8] what teeth are composed of

the jaw bones lower jaw bones: mandibles permanent and deciduous (D), nonpermanent "milk" teeth (laita vaan jos löytyy d-hampaita)

right and left sides are always symmetrical, denoted simply L or R or Lt or Rt or left or right. left is left looking from the animal, not the observers perspective Identicality also causes that sometimes tooth fossils are misidentified to the wrong side and corrected (ei lähteestä vaan nähty datasta koska l ja r on sutattu aika monta kertaa ja vaihettu

four classes, front to back: three incisors (I), one canine (C), four premolars (P), three molars (M). top bottom left right. top/bottom noting upper jaw as superscript lower jaw as lower script, purpose: incisor -¿ catching, canine -¿ stabbing / killing prey, molars are for chewing. premolars are bit like canines bit like molars, function varies lot between taxa including holding, cutting and chewing. also form and number of each present changes between taxa. sometimes lower jaw as line on top and upper jaw as line on bottom, sometimes both are used: upper script number with line on bottom. Line is "the other jaw" if there are less of a type of teeth eg two premolars, they might be no 1 and 2 or no 3 and 4

4 Deep Neural Networks for Optical Character Recognition

separate problems: character classification (easy, kNN, SVM), reading variable-length text (harder) [14] introduce the problem of ocr, example: fossil catalogue

4.1 Deep Neural Networks

- neurons and activation functions. maybe examples of activation functions: relu, sigmoid, softmax

what is a neural net weights in layers: floating point numbers, grouped in groups activations: connections between weights, nonlinear scalar to scalar functions

- feed forward network computes output from input with the feed forward. you have an input, bunch of numbers then, you compute a linear combination and pass that through an activation function

$$h_d = a\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} h_{d-1}\right] \tag{1}$$

hd, hd-1 are hidden unit values. d is layer index. thetas are weights D is width of previous layer [18]

a is any nonlinear funtion, simplest is the rectified linear unit relu, x=x if x¿0, x=0 if x¡0. hidden units are taken as inputs to next layer and next and next. thetas = weights largest, deepest: hundreds of layers, hundreds of millions of weights then last hidden unit output it the output of the network. tadaa.

- universal function approximator the theory of the universal approximation capacity: this algebraic construct, given correct weights, activation functions and structure, could approximate any mapping from input to output. note: input/output dimension can be anything

examples from ocr relevant in this case input is always image ie 2d matrix if grayscale or 3d tensor if rgb image.

classification problem, softmax activation image of tooth sample (letter+number) to four classes, M P I or C. input: image, output: four probabilities that sum to 1, probability this is a m p i or c. sum to 1 is achieved with softmax. output is largest probability.

multilabel classification problem: image of tooth sample. output: first MIPC, second if it is upper or lower jaw output can be two arrays, one like before, other a 0-1 probability for upper jaw. upper if this ¿ 0.5

present sequence to sequence learning: more complex case: image of sentence output: text on image, variable length. here output layer is a more complex structure of probabilities for each position in the result sequence.

insert image example: inputs and outputs of different learnable functions with neural networks

NEW: universal approximation theorem is tighter! it states one layer fc network should be able to approximate any function, so no need even to have the optimal architecture. architectures are for easing practical computation matters

4.2 Training neural networks

u have encoded the structure, starting weights. data, input/output pairs. training = process that adjusts weights so that network becomes a good approximator

1. take part of data as training data 3 divide to batches, common batch sizes are exponents of 2, 2-32 usually 4. pass a batch through the network. get output 5. pass output and correct to loss function: map from output, correct to scalar, 0 is good, large value bad. 6. Compute loss function gradient with respect to network weights using automatic differentiation. algorithm to get this in code is called backpropagation because it moves backward in the network 7. gradient informs how to adjust weights so that loss should decrease. adjust weights in this direction by preset amount called learning rate. Optimizers are algorithms that determine the specifics of "moving in the direction of decreasing loss" most common stochastic gradient descent (inserts randomness to movement) and adam (uses previous iteration movements known as momentum in process to make movements more smooth) 8. run batches until out of data = epoch. run many epochs, stop according to stopping condition that is known to be a state when the network weights are good. goal: reach global minimum of loss function, difficult! would be perfect approximation 8. test on unseen test data to see generalization performance

lots of details on this process, rest is about that

4.2.1 Loss functions

more specific: how do you map output and label to scalar value describing how good the output was? ocr point of view

Loss function is a function from model predictions and ground truth labels that describes with a single scalar value how good the match was, low number describing a good match [18]. These functions are constructed to be equivalent with maximum likelihood solution, think the model would output a conditional distribution of outputs, p(y—x). each ground truth label in the training set should have a high probability in this distribution. Product of all these probabilities is called likelihood. Find parameters that maximize the likelihood of the training data set. Loss functions are derived so that parameters bringing loss to zero is equivalent to the parameters with maximum likelihood. Derivations are out of scope.

- cross-entropy loss kullback-leibler divergence of correct conditional probability and conditional probability parametrized by current model parameters. (show formula, 5.27), correct is not dependent on parameters so is omitted. show 5.29, what is left from that (until here from [18])

then: how cross-entropy loss is computed used for classification problems. eg. is this letter in this image an 'a' or a 'b'. correct probabilities eg .1 and .9 for a or b. model says .2 and .8. discretisized cross entropy computes it as .2*log.1+.8*log.9, log in base 2.

word detection models: have a predefined vocabulary, layer for probability of each word. loss is cross entropy for these probabilities compared to target probability distribution, where correct word has probability 1 and all other have probability 0

- CTC loss

maybe

4.2.2 Evaluating model performance

common in character recognition: accuracy, f1 sometimes also used explain these and differences

accuracy: correct out of those classified

why one uses precision and recall instead of accuracy sometimes simple accuracy is not a good measure, eg cancer screenings: 1 in 500 has cancer. then a dummy saying everyone is healthy would be correct 499/500 of the time, accuracy 99,8%. measurements anyhow relevant would be 99.8-100, inconvenient range.

precision: fraction of positives identified. dummy would have precision 0 recall: out of those noted as positives, fraction of correct ones. dummy would have 0 too f1 is an average to lump these together in one number

going forward: only consider simple accuracy scores as none of the classes are very rare in the fossil case.

4.3 Architectures

different ways of constructing layers, makes model pay more attention to desired things and reduces parameter count from fully connected layers, ie. encode priors [10]

relevant here: present the ImageNet competition that has initiated many new architectures.

4.3.1 Convolutional layers

encode a prior reduces the need to learn parameters. limit is cpu resources and data so given data and cpu, get as good model as you can requires constraining the problem by making assumptions [10]

prior encoded: move image a bit and it is still an image of the same object (translation invariance) and nearby pixels are usually like each other (have a statistical relationship) fully connected nets do not consider input value positions in input vector, how near or far from each other they are, in any way. [18]

practical use: less parameters required -¿ less computational complexity

convolution (cross-correlation) you have input and kernel. input: image matrix (2d) kernel: 2n+1 sized matrix (uneven to make it center around the input pixel processed) kernel slides like a sliding window through the input. for each middle position of the kernel: result is the dot product of the pixels in equal positions, eq 10.6 if the image is rgb, kernel is 3d with depth 3. (bike fig 10.10 here)

aspects: borders, channels, convolution size, dilation and stride default variant reduces size because border pixels cannot be used to combat that: zero or reflective, padding to keep output and input sizes equal, or valid convolution: dont compute positions where there are not enough previous inputs to use the whole kernel channels: run many convolutions in parallel -¿ more convolution kernel weights to learn size stride and dilation, figure 10.3 size = kernel size stride = step between kernel computation dilation = kernel is interleaved with zeros, large region but less weights stride dilation size fig here

usually conv layer followed by pooling layer pooling usually networks are constructed so that layer by layer layer width decreases and has more depth (more channels, like red green and blue but not with this meaning) changing size more than by the few pixels present at image edges: max pooling. take maximum of 2x2 area, collect these values as the output activations. other, less popular variants: mean and average pooling

historical scetch: alexnet [10] brought this to mainstream in 2012, imagenet 2014 saw GoogleLeNet [25] and VGG [23]. these highlighted because the architectures are utilized in handwritten character classification. alexnet, top5 error: 17% best so far whats new regularization with dropout, large model, large conv kernels (11x11-3x3 kernels), relu instead of previously popular tanhs sped up training, multiple GPUs used when training, normalization of convolution results, overlapping pooling (areas of pooling for each pixel overlap) impact: brought deep learning to center of ml research, deep nets seem to outperform humans in feature engineering.

googlelenet [25] won 2014 with top 5 error rate 6,75%. method: fixed number of allowed multiple-adds in forward pass, used inception (named after we need to go deeper meme) layer: 1x1 3x3 and 5x5 sparse convolutional kernels, also sparse fully connected layers ie not all weights are connected to save compute. Also inspired by biological visual systems. allowed for deeper and wider network, turned out to be great in imagenet and object detection.

vgg [23]. achieved 6,8% with imagenet 2014 postsubmission, submitted 7,3% thus second in competition. method: simple 3x3 convolutions, tested different depths. best depth 19. new that previous best models did big kernels and shallower nets. also good model because of the simplicity + the paper included in appendix that demonstrated vgg as a feature extractor to be used in transfer learning.

4.3.2 Transformers

introduced by [28]

- self-attention - multi-head self-attention - tokenizing and the cls token

historical scetch: the vision transformer [4] outperformed many cnns in imagenet score when introduced and changed course of image classification research toward transformers

4.3.3 Autoencoders

- encoder/decoder

4.4 Techniques and heuristics for improving performance

data augmentation

dropout regularization (alexnet uses so need to intruduce)

4.5 Transfer learning

relation to the coarse level training loop: how to initialize model parameters this sect from [16]. they initially formalized the problem and uniformized the terminology

basics: what it is, basic premise: if you start out from a parameter configuration that solves a related task well, there is lesser need to train to solve the new problem [16]. why: save compute (imagenet models vgg lenet alexnet training times 5 days to 3 weeks even when using GPUs [23]) and data labeling (usual constraints in ml model building [9]) also why: training many parameters on a dataset overfits model to the dataset, so freezing layers prevents overfitting [25], optimization idea: using pretraining that makes sense, you start from a parameter configuration in a 'basin of attraction' of a good minimum [5], ie a place where gradients point toward a good local minimum training is strongly influenced by early examples so pretraining prevents overfitting to the supervised task get sgd trapped in a parameter space that is better so resulst is better even when target data is abundant [5]

inductive transfer, transductive transfer, instance transfer, relational knowledge transfer task (inductive) transfer: tasks differ, data same or different task transfer relatedness + the more related the better results negative transfer ie things made worse if things differ too much. super important to consider the distance of transfer ie how similar the tasks are broad methods for inductive transfer: feature representation transfer, parameter transfer; priors or hyperparameters of model are assumed to be shared, instance transfer: reuse source data domain (transductive) transfer (only data differs) central idea in all of these: model first n-1 layers is a feature extractor seeing model as model doing the feature engineering: all but last layer are like feature extraction that allows a linear model to be fit from features to output. first layers learn lower level features, higher up more high level [5], so how much to freeze is about how high level features are usable relational-knowledge-transfer: when data is not i.i.d (not considered here)

my task. source task is image classification with different data than mine inductive transfer where data and task differ. i will test both only feature representation transfer (re-search for optimal hyperparameters) and parameter transfer (use hyperparameters used in source model training) different countries mark tooth fossil in different notation, so fine-tune to adjust

basically nowadays it is usually insensible to ever train from scratch [24] therefore this work definitely transfer learns. also previous work that does not transfer learn is dismissed, as it does not inform this case where target data is scarce

4.5.1 Foundation models

Idea: since transfer learning source task performance is not important, target task is the main point [16], you can come up with some nonsensical task such as map image to the same image to get unsupervised task, train massively on massive compute, and use the model for varieties of tasks. idea: do as little as possible in the target domain because labeling is the expensive thing [9], this helps because unsupervised or selfsupervised allows however many labeled samples one wants.

come up with task with no value in itself but learns useful features. eg identity mapping in autoencoders, mask a part of an image and fill it in, get in a pair of images and say whether they are a transformation of one another. save the pretrained model, can be used with transfer learning in manymany tasks [5]: why unsupervised pretraining works so well analytical analysis: seems to act as a regularizer generalization is better: intuition, model has seen a much larger variety of data

well this is used in ocr as well.

5 Related work

Search strategy: few seed papers and snowball search. Related conferences: Frontiers in Handwriting Recognition

Notes on choices: there is math OCR and music OCR, but they use large datasets and no transfer learning, they dont suffer from the limited target data problem. Also, the problem domain is too different from my problem to be informative.

5.1 Approaches to digitization of handwritten fossil data

[27] reviews various digital paleo data portals. encourages further digitization but does not really mention how one should go about doing that

[13] review of digitizing fossil data. only considers digitizing physical bone pieces to various 3d images. does not take any stand on digitizing handwritten notes on the samples

[7] does identify my problem here: most fossil data is in handwritten paper format with many things already known such as taxon. gives suggestions on data format and management but mostly assuming that there is a human transcriber. presents ocr reading as a possibly one day possible option. closest to automation known in this work was making imaging of physical samples more

efficient with a conveyor belt system, which has nothing to do with handwriting ocr. Notes that especially cleaning of automated ocr read data is a hard problem.

only work doing the exact same thing: [21], but quality of the work is very poor, only presents rudimentary very basic general aspects of image processing such as canny edge detection or contour detection from characters, which is very very far away from working ocr. the paper lists absolutely no results but is an evidence that someone has somewhere at least attempted this.

So: to the best of my knowledge there is no other work successfully reading handwritten fossil scans or cleaning already automatically read systems outside of 2024 spring data science project for KNM, for which this is the continuation. therefore also here i will do the simplest things first, since it is a good idea to start with simple and once successful, proceed to harder problems that would digitize and clean more data.

5.2 Approaches to character recognition with small target domain datasets

method: snowball search. for a paper to be accepted for use here, i set up these requirements: lists best accuracy clearly (percent, which dataset, number of classes) specifies the base model used (architecture and source domain dataset) specifies how transfer learning was conducted (not just used transfer learning) then reputable venue and acceptable quality of text and presentation, super subpar text was a frequent reason to skip a paper, they also often lacked relevant details

6 Experimental setup

6.1 Data description

has been done by different annotators, no logs on who logged what, everyone had a bit different style of notating. also no clearly defined standard for notating specimens. so might be that actual data used will have characters or words not present in any data, causing errors.

Identicality also causes that sometimes tooth fossils are misidentified to the wrong side, seen in data with smudged over l's and r's

Catalogues have lines between entries. challenge for the model to not mark these as underlined in these cases the line is long and spans the entire image, so it should be distinguishable from a single underlined character.

difference to other domains of big data: each fossil sample is expensive and human laborous to obtain, thus "big" sample is small compared to other ml applications, 1000 is a lot 10k huge.

6.1.1 Notes on creating the dataset

Hand-labeled

Data was extracted from scans by getting bounding boxes from Azure Vision API, finding the correct column (nature of specimen or element), and cropping the image according to bounding boxes.

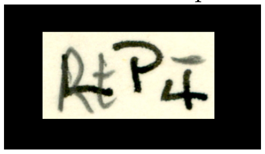
Non-tooth samples were not discarded since they contain bone fossil related words and good samples of the handwriting style of this dataset.

smudged-over "L" was labeled as "R", and other way around: it seems that later someone found it was the opposite side after all. Hope of this is that the model would learn to map "messy L" as "R". snudged "left" or "right" was not noted as the opposite as there were too few such samples.

Superscript seems much more rare than lower script

Data was labeled not by individual characters but as full tooth descriptions to preserve context where tooth special characters are more likely to occur

Some have been corrected by writing on top and thus are very hard even for humans to read, this is also an example of smudged-over correction:



6.1.2 Unicode characters used for data labeling

explain: unicode has graphenes with code points. eg a is one graphene one code point, à is one graphene two code points (dot on top and the letter). the top thing -like characters will be called "modifiers".

markings contain letters and numbers with no line, line on top or line at the bottom. Each character can be lower- or upper script. The modifiers used are: macron with lower (Ā) and upper variant.

Unicode [26] has characters that are for example upper script, but these were not used for two reasons:

- lower and upper script character set is incomplete for this purpose (eg 3 with upper macron and lower script needed)

- from the model perspective 3 and $_3$ are no more similar than A and B, however, three combined with lower script modifier and 3 with upper script modifier all contain the same unicode character 3 with only the modifier changing. The problem here is that there is no lower or upper case modifiers in unicode. Therefore, the caron (Ǎ) was chosen as the lower script modifier, and the circumflex accent (Â) as upper script. These were chosen since the arrow-like modifier pointing up or down is maybe the most logical placeholder for the missing modifier. More traditional workarounds of missing upper or lower script, the underscore "_" and separate caret character "ˆ" were not used to keep one unicode graphene represent one character on the page. Also on the other hand using one modifier for all lowercase characters allows the model to understand that there is a similarity between all lowercase characters. The intention is that one idea about a character is encoded as one code point, so that the model can learn the mapping from the image of the character to the code point combination (until now already in thesis text) —-

also: some annotators used / instead of line. left to / -¿ upper, right -¿ lower. unknown/unsure noted with x with macrons on top/bottom annotate with up/down macron

if model is toothornot classifier + tooth reader -¿ remove fractions notes

6.2 Data preprocessing

grayscaling, flipping to back background white foreground

I should especially do random cropping since that occurs in real dataset!

I sould do some border contrast decrease center contrast increase because border noise is present and usually less relevant

all these: test first without, then add preprocessing step, then re-test

i am able to get labeled MPIC data by getting azure outputs where the letter is mpi or c segmenting out the letter: does not make sense since the premise of deep learning is to to feature extraction better than a human would do. to segment out the letter you would need to know which area is the letter and to know that you need to know what letter there is which is the problem in the first place, so unsolvable egg hen problem

6.3 Methods: base models and transfer learning tehniques

6.3.1 Problem formulation

different types of defining inputs and outputs, discussion on what to choose

the sequence or character per character recognition question: sequence is a mapping from image to variable-length phrase character per character approach, inspired by [31]: train a classifier: is this word a tooth or not? then give non-tooth to the untuned trocr, which works very well. Then few shot transfer learn a classifier from an image with only a tooth marking (letter and one number) to tooth. Possibly extend classifier to be able to recognize multiple teeth (eg M1-3). Target could be multivariate: first would be tooth (which i1-3,c,p1-4,m1-3), second would be jaw (upper, lower, unknown), third side (left, right, unknown). Separating 'l/r/lt' from the letter+number tooth notation is fairly trivial: noncursive handwriting can be separated by finding a vertical line where there is no black. Image processing: convert to black and white, then find x coordinate with no black and split there.

sequence benefits: adaptable to many kinds and lengths of input, possible to get good inferences for surprising marking styles sequence bad sides: finetuning just one layer on 80 training images for two epochs took about 15 minutes -¿ all hyperparameter optimization etc is out of question with this heavy training. Also: not domain adaptation (adapting same task to different dataset), but task adaptation ie. the target set of characters has changed. Encoding this to the large encoder decoder transformers would require rewriting parts of the preprocessor and model which is too complex given the level of this work.

character by character benefits: feasible given available data and computing resources possible to encode the classes (ie, teeth) classifying characters has been essentially solved, easy problem also classifying to tooth or not tooth should be easy –¿ focus on model ensemble with tooth or not classifier + trocr + tooth classifier

multilabel class or classifier chains: multilabel is difficult because the basic case has just set of labels and you can have any number of any of them. we have 4 types of classes with dependencies (no 4th canine) so chains make sense, as constrained multilabel classification with class groups is a complex, nonstandard construct [30]. so, mini model approach

1. tooth or not: azure output text matches regexp letter and number or 'c' or 'C' -¿ 1/0 train using MPIC images 2. MPI classifier: tooth type (regexp already says c is c), upper lower classifier: 1/0 3. get number (1-4) from azure output, is generally correct by visual inspection

not inspired from any paper but my own idea, felt like a sensible way to formulate the problem, so we have for all models a well-solved well-defined basic classification problem.

reason for this choice: it encodes most prior knowledge in the output structure, use as much from azure as possible

priors: there is also the azure output for the tooth. it is likely to be very good but not constrained to the MPIC case. include the azure label as one input variable why we went for the simplest problem out there because the part of data that can be digitized with the tightest constraints and the simplest models are most accurate. Do that first, deal with the remaining later

6.3.2 Base model selection

initial sequence learning attempts done with [11], proof of concept so no real exhaustive comparison literature review

Chosen public dataset was EMNIST [2] because it is the closest to my problem: a large dataset of labeled letters and numbers.

According to [1], this model was the best: [22], was chosen as the base model. they trained on emnist-balanced, so no difference between upper -and lower case letters. not a problem for us since in the handwritten catalogues, upper or lowercase was not used to distinguish upper and lower jaw

mnist and real world fundamental difference: mnist is perfect world, real images vary much more [10] therefore: not e(mnist) as source problem, but imagenet classification.

three lines: imagenet, unsupervised pretraining (denoising autoencoder?), and MNIST/EMNIST (my hypothesis: transfer distance is shorter could be an advantage) also the fact that the related work was not that high quality, no readable code published so i don't trust them without question...

6.3.3 Transfer learning method selection

Priors. base model already knows the output should be a word, eg "jdaslkjflkds" is a highly unlikely correct answer. Bone notation has a very small subset of possible english words, eg. the word "beach ball" cannot ever be a correct answer for a reading –¿ these do not apply if we do character classification

some papers did some first layer unfreezing, or middle layer unfreezing ie something else but not freezing last layers. dismiss that because they did not state why they did that. from the learn higher and higher level features point of view having an not frozen layer and then after that a frozen one does not make any sense; the later layer is based on the previous. so never in experiments have a frozen layer below a layer that is frozen

7 Results and discussion

table of results could contain: rows: base models tested columns: transfer learning method

transfer learning methods could include

just training further (ie baseline, dont do anything special)

train the last layer (inspo from [31], test hypothesis of reusing feature extractor) what else? find from literature!

train unsupervised autoencoder on the target data (reduces domain transfer distance), transfer to supervised classification task

cleaning data from original format with gaussian blur and otsu thresholding surprisingly did not change results much

augmentation to balance classes resulted in a 10% increase in tooth type classification accuracy using a simple translation. translation also makes sense since that is the main variable changing between images, zoom or rotation changing is more rare as these are scans.

initial phase experiments: mnist transfer to mpi since it was thought to be a similar problem. however like noted by [10], real world image classification is much more complex than digit classification as the benchmark images are ideal cases with eg perfectly even background. Therefore later phases experimented with transfer from imagenet classifiers since that is what most of papers in related work also did.

8 Conclusions

how this could be continued main problem in working with azure bounding boxes: incorrect identification of word bounding boxes. encoding prior knowledge should work better: M1-3 is a word for instance, azure did stuff like m, 1-, 3 from computer vision lecture 11 [20] object bounding box detection requires higher resolution images and ground truth labels -¿ labeling effort is big, compute needed. one idea is to take premade bounding boxes by generalist ocr word detectors and for example infer they are correct by matching with a vocabulary of known correct words to exist in fossil catalogues.

I quickly tested a state of the art object detection model the yolo10, variant of popular object detector yolo[19] untuned, on huggingface api, the yolo10 [15]. The performance was poor, see fig, probably since object detection usually does 3d object detection in a 3d scene, so one should either fine tune or find a ocr bounding box detector. then there is open-vocabulary object detector which you can tell what objects are present [29]. we could just say there are teeth and words. then also a very recent publication on detecting text on images [12]. benefit of these is that they are fine-tunable which the commercial azure api is not. then also detectron [3] from meta for object detection one could also run more sophisticated tooth or not classification on azure output, which could work very well but relying on a paid service is not ideal for open access solutions.

ways to improve the result of image classification obtained here test svm after feature extraction instead of the dense layers + softmax. My most recent model was from (insert year), since then imagenet classification has advanced, so try using a newer base model. these can be harder to obtain since they are not as established, I could just fetch models from torchvision.models, newer ones are not present there. with a good object detector probably we can get to more variaty of tooth marking images (maybe a fig of examples?) so the classification task would become more complicated. Still, the downstream tooth marking image to tooth task is a relatively easy classification problem comparing to eg imagenet, so the main challenge is definitely finding words and classifying them to tooth or not.

this was a very quick literature search and practical test, so proper thorough work should be done to verify if this speculation is correct or not.

References

[1] A. Baldominos, Y. Saez, and P. Isasi. "A survey of handwritten character recognition with mnist and emnist". In: *Applied Sciences* 9.15 (2019), p. 3169.

[2] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. "EMNIST: Extending MNIST to handwritten letters". In: *2017 international joint conference on neural networks (IJCNN)*. IEEE. 2017, pp. 2921–2926.

[3] *Detectron*. URL: `https://ai.meta.com/tools/detectron` (visited on 10/11/2024).

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. *An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale*. June 3, 2021. arXiv: `2010.11929 [cs]`. URL: `http://arxiv.org/abs/2010.11929` (visited on 09/26/2024). Pre-published.

[5] D. Erhan, A. Courville, Y. Bengio, and P. Vincent. “Why Does Unsupervised Pre-training Help Deep Learning?” In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, Mar. 31, 2010, pp. 201–208. URL: `https://proceedings.mlr.press/v9/erhan10a.html` (visited on 09/26/2024).

[6] J. T. Faith and R. L. Lyman. *Paleozoology and Paleoenvironments: Fundamentals, Assumptions, Techniques*. Cambridge University Press, 2019.

[7] Q. Groom, M. Dillen, H. Hardy, S. Phillips, L. Willemse, and Z. Wu. “Improved Standardization of Transcribed Digital Specimen Data”. In: *Database* 2019 (Jan. 1, 2019), baz129. ISSN: 1758-0463. DOI: `10.1093/database/baz129`. URL: `https://academic.oup.com/database/article/doi/10.1093/database/baz129/5670756` (visited on 09/30/2024).

[8] S. Hillson. “Tooth Form in Mammals”. In: *Teeth*. Cambridge Manuals in Archaeology. Cambridge University Press, 2005, pp. 7–145.

[9] C. Huyen. *Designing machine learning systems.* ” O’Reilly Media, Inc.”, 2022.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: `https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html` (visited on 10/02/2024).

[11] M. Li, T. Lv, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei. *TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models*. 2021. arXiv: `2109.10282 [cs.CL]`.

[12] S. Long, S. Qin, Y. Fujii, A. Bissacco, and M. Raptis. “Hierarchical Text Spotter for Joint Text Spotting and Layout Analysis”. In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Waikoloa, HI, USA: IEEE, Jan. 3, 2024, pp. 892–902. ISBN: 9798350318920. DOI: `10.1109/WACV57701.2024.00095`. URL: `https://ieeexplore.ieee.org/document/10483893/` (visited on 10/11/2024).

[13] H. Mallison. “Digitizing Methods for Paleontology: Applications, Benefits and Limitations”. In: *Computational Paleontology*. Ed. by A. M. Elewa. Berlin, Heidelberg: Springer, 2011, pp. 7–43. ISBN: 978-3-642-16271-8. DOI: `10.1007/978-3-642-16271-8_2`. URL: `https://doi.org/10.1007/978-3-642-16271-8_2` (visited on 09/30/2024).

[14] J. Memon, M. Sami, R. A. Khan, and M. Uddin. “Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)”. In: *IEEE Access* 8 (2020), pp. 142642–142668. DOI: `10.1109/ACCESS.2020.3012542`.

[15] *Omoured/YOLOv10-Document-Layout-Analysis · Hugging Face*. Jan. 25, 2023. URL: `https://huggingface.co/omoured/YOLOv10-Document-Layout-Analysis` (visited on 10/11/2024).

[16] S. J. Pan and Q. Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359.

[17] M. M. Phillips, K.-C. Cheng, and A. Hotz. *A Museum Overflowing With Prehistoric Treasures Races to Save Itself*. The Wall Street Journal. Oct. 12, 2024. URL: `https://www.wsj.com/world/africa/nairobi-national-museum-natural-history-leakey-832f0262` (visited on 10/15/2024).

[18] S. J. Prince. *Understanding Deep Learning.* The MIT Press, 2023. URL: `http://udlbook.com`.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, June 2016, pp. 779–788. ISBN: 978-1-4673-8851-1. DOI: `10.1109/CVPR.2016.91`. URL: `http://ieeexplore.ieee.org/document/7780460/` (visited on 10/11/2024).

[20] L. Ruotsalainen. *Computer Vision Applications with CNNs.* Lecture for the course "Computer Vision" presented at the University of Helsinki, Faculty of Science. Oct. 2024.

[21] S. Shanmugavel, J. Kannan, A. Vaithilingam Sudhakar, and . . "Handwritten Optical Character Extraction and Recognition from Catalogue Sheets". In: *International Journal of Engineering & Technology* 7.4.5 (Sept. 22, 2018), p. 36. ISSN: 2227-524X. DOI: `10.14419/ijet.v7i4.5.20005`. URL: `https://www.sciencepubco.com/index.php/ijet/article/view/20005` (visited on 09/30/2024).

[22] A. Shawon, M. Jamil-Ur Rahman, F. Mahmud, and M. Arefin Zaman. "Bangla Handwritten Digit Recognition Using Deep CNN for Large and Unbiased Dataset". In: *2018 International Conference on Bangla Speech and Language Processing (ICBSLP).* 2018, pp. 1–6. DOI: `10.1109/ICBSLP.2018.8554900`.

[23] K. Simonyan and A. Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv.org. Sept. 4, 2014. URL: `https://arxiv.org/abs/1409.1556v6` (visited on 10/04/2024).

[24] C. C. N. N. f. V. R. Stanford University. *Transfer Learning.* `https://cs231n.github.io/transfer-learning/`. Accessed: 2024-10-16. 2024.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going Deeper With Convolutions". In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, pp. 1–9. URL: `https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html` (visited on 10/07/2024).

[26] The Unicode Consortium. *The Unicode Standard.* `https://home.unicode.org/`. [Accessed: 2024-09-04]. 2024.

[27] M. D. Uhen, A. D. Barnosky, B. Bills, J. Blois, M. T. Carrano, M. A. Carrasco, G. M. Erickson, J. T. Eronen, M. Fortelius, R. W. Graham, E. C. Grimm, M. A. O'Leary, A. Mast, W. H. Piel, P. D. Polly, and L. K. Säilä. "From Card Catalogs to Computers: Databases in Vertebrate Paleontology". In: *Journal of Vertebrate Paleontology* 33.1 (Jan. 1, 2013), pp. 13–28. ISSN: 0272-4634. DOI: `10.1080/02724634.2012.716114`. URL: `https://doi.org/10.1080/02724634.2012.716114` (visited on 09/24/2024).

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention Is All You Need". In: ().

[29] *YOLO-World: Real-Time Open-Vocabulary Object Detection.* ar5iv. URL: `https://ar5iv.labs.arxiv.org/html/2401.17270` (visited on 10/11/2024).

[30] M.-L. Zhang and Z.-H. Zhou. "A Review on Multi-Label Learning Algorithms". In: *IEEE Transactions on Knowledge and Data Engineering* 26.8 (2014), pp. 1819–1837. DOI: `10.1109/TKDE.2013.39`.

[31] G. Zhao, W. Wang, X. Wang, X. Bao, H. Li, and M. Liu. “Incremental Recognition of Multi-Style Tibetan Character Based on Transfer Learning”. In: *IEEE Access* 12 (2024), pp. 44190–44206. DOI: 10.1109/ACCESS.2024.3381039.