# Molecular smell prediction using deep neural network ensemble

Cas van Boekholdt
STUDENT NUMBER: 2047042

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN DATA SCIENCE & SOCIETY
DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE
SCHOOL OF HUMANITIES AND DIGITAL SCIENCES
TILBURG UNIVERSITY

Thesis committee:

Dr. Giacomo Spigler
Dr. Juan Sebastian Olier Jauregui

Tilburg University
School of Humanities and Digital Sciences
Department of Cognitive Science & Artificial Intelligence
Tilburg, The Netherlands
January 2021

# Contents

# Molecular smell prediction using deep neural network ensemble

Cas van Boekholdt

*Studying the relationship between odorants' molecular structure and human-perceived olfaction is traditionally a task in biochemistry and cognitive neuroscience, dating back decades. Developments in machine learning and cheminformatics allow for novel approaches to the task of predicting human olfaction, and successful prediction produces societal implications, including applications of electronic noses and reduction of environmental impacts as a result of harvesting natural odorants. This research revolves around the application of a neural network ensemble to predict molecules' smells using an expert-labeled dataset. This ensemble's prediction results show similar prediction performance compared to the current state of the art in this task with minimal biochemical knowledge application. Furthermore, a novel web-scraping approach is proposed utilizing the relationship between olfaction and the more literature-prevalent mechanism of human taste with the application of automated natural language processing. This novel approach exceeds the state of the art prediction performance when combined with the neural network ensemble.*

## 1. Introduction

Replications of the human senses are extensively addressed in machine learning literature, with significant recent vision and hearing modeling developments (Kheradpisheh et al. 2016; Kell et al. 2018). Although these two senses are the most obvious to model, as data from these senses can be easily digitalized using cameras and microphones, modeling the sense of smell is far underrepresented in comparison, while great societal benefits could arise from modeling the human olfactory sense. Directly predicting human perception of a molecule's olfaction would make it possible to discover and develop new synthetic odorants, which would reduce the harmful environmental impact that is the result of harvesting natural odorants. Various applications could also arise from smell prediction in the context of sophisticated electronic noses, which are devices capable of classifying odorants from their physical structure (Gardner and Bartlett 1994). Such devices are currently simple but could be improved significantly if smell prediction is possible from molecular structure.

This thesis's research domain can be classified into quantitative structure-odor relationship (QSOR) modeling, a sub-domain of quantitative structure-activity relationship (QSAR) modeling. In this domain, various efforts have been made using either traditional machine learning methods or methods relying heavily on expert domain knowledge. Few efforts have been made to solve this problem using deep learning approaches with minimal application of domain knowledge, which is what reaped breakthrough results in vision and hearing modeling. This thesis's contribution to the QSOR research domain will explore different molecular input representations requiring little application of domain knowledge to accurately predict human olfactory percep-

tion using deep learning approaches. The following research question was formulated to guide this exploration:

*RQ: Can modern deep learning algorithms using standard molecular input representations perform smell prediction for molecules on par with state of the art quantitative structure-odor relationship models?*

In order to answer this research question, various subquestions were formulated. First, the research revolves around standard molecular input representation. Therefore, it is essential first to explore what types of standard molecular input representations are available, their benefits and drawbacks, and how they perform in olfaction modeling. Thus, the following subquestion was formulated:

*SQ1: Which standard molecular input representations yield the strongest predictive value for olfaction modeling?*

Possible input representations of SMILES molecules are fingerprint embeddings, graph tensors and images. With different input representations come different advantages and drawbacks. An interesting subject to explore is whether multiple input representations could be utilized simultaneously to improve overall prediction performance. These input representations and the combination thereof will be further discussed in section 3. The following subquestion was formulated to explore the combination of input representations:

*SQ2: Can the varying benefits of multiple molecular input representations be utilized conjointly to improve smell prediction performance to state of the art level?*

Data availability on molecule odors is limited, while the similar mechanism of human taste is better represented in literature and data availability is, therefore, greater in this domain. An interesting factor to explore is the relationship between labels used to describe taste and smell and the possibility of utilizing this relationship to improve smell prediction. The following subquestion was formulated to address this aspect:

*SQ3: Can the overlap between labels used to describe molecules' taste and smell be exploited to improve olfaction prediction performance?*

## 2. Related Work

This section serves to introduce the related work in the scientific literature. The background of machine learning for molecules will be introduced, with a focus on molecular smell prediction particularly. While a full history of deep learning falls outside the scope of this thesis, developments relevant to this research will be briefly introduced.

### 2.1 Machine learning for molecules

Machine learning for molecules has been increasingly maturing in recent years due to improved methods and the presence of larger datasets, allowing machine learning algorithms to make more accurate predictions about molecular properties (Wu et al. 2018). Any molecule can be represented with simplified molecular-input line-entry specification (SMILES) strings, which are unique representations of molecules (further explained in 3.1). However, prediction methods require further information to learn features from molecules when data is limited (Wu et al. 2018). Many efforts have been made to transform SMILES strings into representations that contain a more robust predictive value.

Chemception is a machine learning framework that transforms SMILES strings into square images to train a deep neural network to predict molecules' toxicity (Goh et al. 2017). The approach shows the ability to perform prediction for molecules through

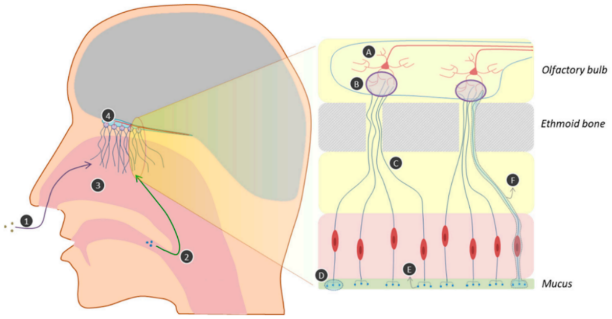
machine learning with performance on par with more classic QSAR approaches that require the extensive application of domain knowledge.

Graph neural networks are novel architectures that can learn features directly from graphs, which are highly suitable for machine learning for molecules due to the natural representation of molecular graphs. However, they require the incorporation of symbolic domain knowledge, which could prove a bottleneck to machine learning researchers (Dash, Srinivasan, and Vig 2020).

### 2.2 Molecular smell prediction

The mechanism of olfaction in humans can be simplified to the airflow of odorants binding to receptors, which transduces a signal to the brain, in which the information is processed. The system is visualized in figure 1. Since the transduced signal depends on the odorant molecules' structure, humans' smell perception can be predicted by examining an odorant's molecular structure (Genva et al. 2019). The mechanism of human olfaction is closely related to the taste mechanism in humans, and the perception of taste is, like smell, dependent upon molecular structure (Buck 2006).

**Figure 1**
The mechanism of human olfaction. 1. Orthonasal; 2. retronasal; 3. nasal cavity; 4. olfactory bulb (Genva et al. 2019).




Early applications of electronic noses date back as far as 1961 with simple applications, including volatile gas detectors (Gardner and Bartlett 1994). With increasing data sets and developments in machine learning, such applications are becoming increasingly intelligent. Current electronic nose applications can often detect volatile gases undetectable by human olfaction but can not yet match human olfaction in terms of the diverse circumstances it can perform in (Genva et al. 2019). In a comparative paper (Kell et al. 2018), researchers compared the results from top submissions of the DREAM Olfaction Prediction Challenge, in which participants predicted odor intensity, pleasantness and categorized molecules into a small set of semantic descriptors of smells. Although the results show promising possibilities in this field, the prediction performance on smell categorization is limited by the limited data size, allowing only traditional machine learning approaches such as random forests to be explored.

The current state of the art performance on molecular smell prediction arose from an application of graph neural networks (GNNs) on a novel data set labeled by olfactory experts (Sanchez-Lengeling et al. 2019). The presented model in this research outperforms model architectures, including random forests and K-nearest neighbors, which were the state of the art up until then. The model does not prove to be able to transfer

its learning to other smell prediction tasks and datasets, as the previously mentioned DREAM Olfaction Prediction Challenge (Kell et al. 2018) dataset is tackled. The mean performance is matched, but performance is indistinguishable when taking into account confidence intervals. In this application, molecules are featurized by their constituent atoms, bonds, and connectivities, after which GNN layers transform the features into a fixed-length vector, which is used to train a fully-connected neural network (Sanchez-Lengeling et al. 2019). This particular featurization into graphs is highly customized to the problem and is therefore dependent upon extensive domain knowledge (Dash, Srinivasan, and Vig 2020).

This thesis aims to fill the gap in the research domain between poorly performing classic machine learning approaches and well-performing GNN approaches that require domain knowledge application. The goal is to develop a neural network model that can reap the benefits of multiple input representations from previous literature to predict smells of molecules with the application of only data science researcher-level domain knowledge.

### 2.3 Deep learning

Deep learning allows the learning of data representations with multiple abstraction levels by computational models with multiple processing layers. Developments in deep learning have created breakthroughs in image, video, and audio processing by using a backpropagation algorithm to teach a model to generalize over learned data (Lecun, Bengio, and Hinton 2015). A significant amount of research is being done to improve the performance of deep learning methods on different problems. Within deep learning, some of the most prevalent architectures are fully-connected neural networks (FCNNs, also referred to as dense neural networks), convolutional neural networks (CNNs), and recurrent neural networks (RNNs). Each of these model architectures performs well at specific tasks, and they are hence complementary in their modeling capabilities (Sainath et al. 2015).

Convolutional neural networks have proven to perform well at problems in which the data have spatial relations, such as in image classification (Krizhevsky, Sutskever, and Hinton 2012). In image classification, data augmentation in the form of rotation, cropping, and zooming has proven successful in improving prediction performance by artificially increasing the training dataset's size (Bloice, Stocker, and Holzinger 2017). Transfer learning is a common way of increasing model performance by using a pre-trained model to extract features from data and using these features to train fully-connected neural networks. A typical application of such transfer learning is VGG16 feature extraction (Simonyan and Zisserman 2015).

Neural network ensembles (NNEs) combine the outputs of multiple neural networks, which are trained either individually or conjointly to improve the generalization performance. This approach aims to increase model diversity and allow the use of multiple input representations to a network. Results of such ensembles show great improvement in prediction performance (Li, Wang, and Ding 2018).

### 2.4 Natural language processing

The volume of natural language text in the internet-connected world is abundant, but having large bodies of text does not inherently mean having a large knowledge base, especially when taking into account time limits for text analysis (Chowdhary 2020). Automated natural language processing (NLP) techniques aim to analyze texts

in natural human language effectively and accurately. The goal of NLP is to be able to analyze texts in the same way humans would, with considerably higher efficiency, to reap the advantages of the vast amount of text available.

**2.4.1 Dependency parsing.** Syntactic text parsing based on dependencies across words in a natural language text has become an increasingly popular method in NLP (Kübler, McDonald, and Nivre 2009). The goal of dependency parsing is to analyze which words are dependent upon each other for each sentence in a text. Classically, dependency parsing algorithms are based on transitions or hard-coded grammars. The rise of deep learning has also progressed methods in NLP, and neural network dependency parsers are on the rise, showing superior performance to classical dependency parsing methods (Zhang, Zhao, and Qin 2016).

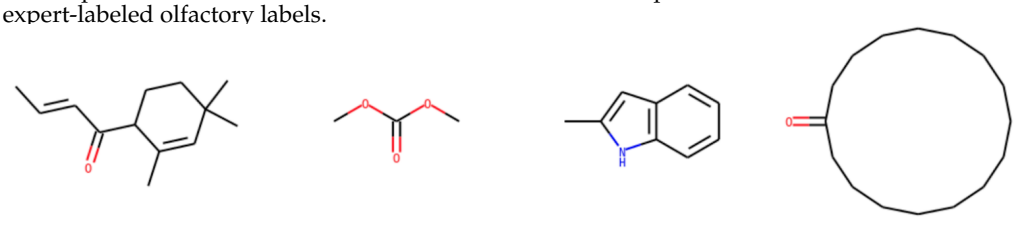
## 3. Experimental Setup

This section will introduce the setup of all experiments carried out to get the insights required to answer the research questions relevant to this study. This includes information on the used data, model architecture, and evaluation. Examples of all methods described in this section can be found in Appendix C: Prediction examples.

### 3.1 Data

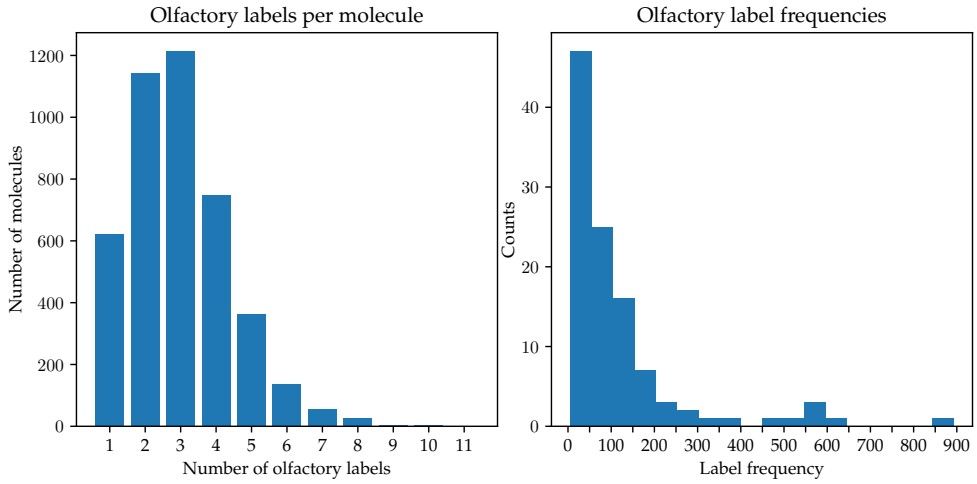
The dataset used in this research was published by crowdsourcing platform AIcrowd for their 'Learning to Smell Challenge' (Mohanty). This set is based on the 'Database of Perfumery Materials & Performance (PMP)' (Boelens) and cleaned by AIcrowd to only include the relevant information of molecules and olfactory labels. The dataset consists of a total of 5,395 molecules, separated into disjoint train/validation and test sets of sizes 4,316 and 1,079, respectively. The train/validation set contains features and labels, and the test set only contains features. Every molecule in the dataset is represented with its respective SMILES representation. SMILES (Simplified Molecular Input Line Entry System) is a molecular notation designed to allow chemical information processing based on a minimal grammar (Weininger 1988). figure 2 shows a depiction of samples in the dataset, with a visualization of molecules and their respective SMILES notation.

**Figure 2**
Visualizations of molecules with their SMILES notation and olfactory labels. From top to bottom, rows represent: Molecule visualization; molecule's SMILES representation; molecule's expert-labeled olfactory labels.



Each molecule in the train/validation set has a selection of 1-11 of the total 109 olfactory labels, thus creating a multi-label classification problem. The most common number of labels per molecule is 3, the least common is 11, and the full distribution of label sizes can be seen in figure 3. The label frequency in this dataset is highly imbalanced, with most labels appearing between 0-50 times and a few most frequent labels appearing over 300 times. See figure 3 for the distribution of label frequencies.

**Figure 3**
Analysis of label distribution. Left figure shows a histogram representation of the number of distinct labels per molecule in the dataset. The right figure shows a histogram representation of the frequency in which the labels are present in the dataset.




### 3.2 Supplementary data collection

Due to the similar mechanism of human olfaction and taste (Buck 2006), descriptors used to describe smell and taste can be similar. As the taste mechanism is more prevalent in academic literature, there is also data available for many molecules on their individual taste. The largest database of molecules, Pubchem, contains information on 109 million molecules (Kim et al. 2019). A large part of this information is structural, including molecular properties such as molecular weight, atom count, and formal charge. All structural information available in the Pubchem database can be accessed efficiently through the Pubchem Rest API. Besides structural information, there is also a large volume of unstructured data including taste information in the form of natural language available for many of the molecules in the Pubchem database. This unstructured data is not available in the Rest API, but only in the web interface. In this research, an attempt is made to structurally extract taste labels for molecules using two different methods: The naive method, and the NLP method. The first step for both methods is to extract the unstructured data from the database entry of each molecule. Since the unstructured data is only available in the web interface, a web scraping technique using Python Selenium is used for data extraction. After data extraction, the distinction is made between the Naive and NLP methods.

**3.2.1 Naive method.** In the Naive method, a simple algorithm checks if the word is present in the extracted text for every one of the 109 possible olfactory labels available in the original dataset. This algorithm then creates a binary vector of size 109, with values being 0 if the respective word is not in the extracted text, and values being 1 if it is in the text. The Naive method is naive because it assumes that if the word is in the text, it will always be a description of a taste, which is an invalid assumption in many cases. The benefit of this method is that false negatives are limited, while the drawback is that there will be a considerable amount of false positives.

**3.2.2 NLP method.** The NLP method is more sophisticated and uses neural network dependency parsing in order to better analyze whether or not a word is used to describe a taste. The parsing model used is extended from the Biaffine Parser (Dozat and Manning 2016) and exploits word embeddings and transformer-provided attentions. The resulting word dependencies are used in an algorithm that again creates a binary vector of size 109. In this case, for a positive label, the word has to be in the extracted text and the dependent word needs to be in a pre-specified list of words including taste, flavour, and flavor. The advantage of the NLP method is that it will limit the number of false positive labels, while the drawback is that it might introduce false negatives by being too strict. The difference between the Naive and NLP method is illustrated with two examples in figure 4

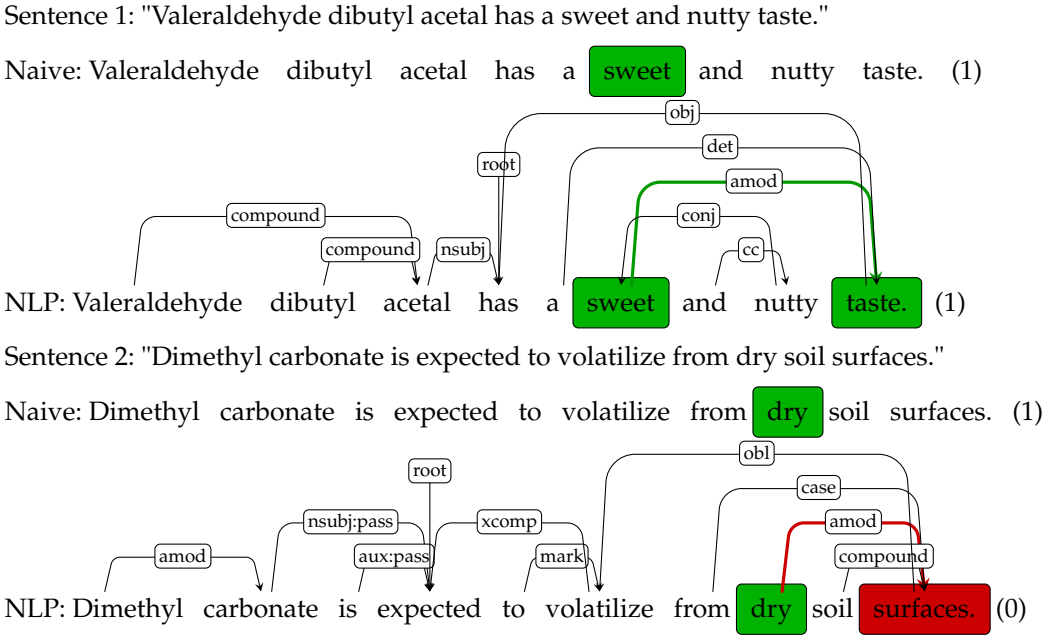
**3.2.3 Supplementary data usage.** The aforementioned methods of automated taste label generation were used to collect information on two sets of molecules. Firstly, the taste labels were generated for all SMILES in the training data set of which olfactory labels are known. This allows the overlap between expert-labeled olfactory labels and automatically labeled taste labels to be gauged. These taste labels were also used to improve the prediction performance of the final model ensemble, as further explained in section 3.3.4. Secondly, the methods of automated taste label generation were used to create an additional dataset of molecules of which the olfactory labels are unknown. The goal of this additional dataset is to see if the overlap between labels used to describe taste and smell can be used to improve model generalization. The additional dataset was used as training data together with the original dataset. For this purpose, a total of 5,000,000 molecules' Pubchem entries were analyzed to attempt to create a novel dataset of SMILES with their respective taste labels.

### 3.3 Feature engineering and model architecture

The SMILES representations present as features in the dataset are not usable as features in a deep learning model. Therefore, the SMILES representations were transformed into multiple input representations, each explained in a separate subparagraph. Each input representation is different in shape, ranging from images to vectors to matrices. Therefore, different model architectures are required for each input representation. Every model in this research is constructed using the Python programming language with Tensorflow and Keras libraries (Abadi et al.). The comparisons of different model architectures for each type of input can be found in Appendix C: Model comparison.

**3.3.1 Chemception.** Chemception is a deep neural network approach to quantitative structure-activity relationship modeling that matches expert-developed models' performance with minimal domain knowledge application (Goh et al. 2017). The Chemception model was developed for a toxicity and activity prediction task, but the preprocessing

**Figure 4**
Two examples of label creation using the Naive and NLP methods on real sentences in the dataset. Numbers in parentheses represent the label for each combination of method and sentence. Sentence 1 shows how a word from the vocabulary is found which is also dependent by adjectival modifier on a word related to taste. Sentence 2 shows how a word from the vocabulary is found, but it is dependent by adjectival modifier on a word unrelated to taste.

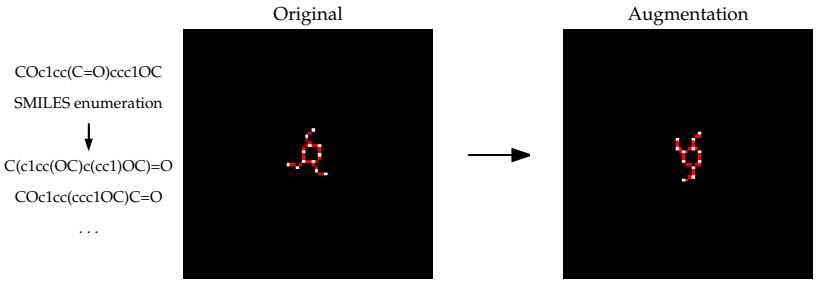



step can be applied to any QSAR task type. Chemception progresses on the advancements made in image classification by representing molecules as 80x80 pixel images and using the Inception-ResNet v2 (Szegedy et al. 2017) neural network architecture for its regression task.

A very common prediction optimizer for image classification tasks is image augmentation. By augmenting images from a training set, the training set size can be artificially inflated, often leading to better model generalization (Bloice, Stocker, and Holzinger 2017). The most common image augmentation method uses the Keras ImageDataGenerator to generate new images based on rotations, zooming, shifting, and other augmentations. While this is a well-performing approach for many image classification tasks, such augmentations generally yield an invalid molecule, which means these methods are unfit for this research task. Since SMILES representations are not unique representations of molecules, multiple SMILES can depict the same molecule from a different entry point (Bjerrum 2017). Therefore, instead of augmenting images created with Chemception, SMILES enumeration was applied to the dataset and the Chemception preprocessing was used on the original and augmented SMILES. The result of SMILES enumeration before and after the Chemception preprocessing can be seen in figure 5.

For the classification of the images engineered with chemception preprocessing, two approaches were explored. The first approach trains a convolutional neural network using the images directly as input. The second approach applies transfer learning in the form of feature extraction with the pre-trained VGG16 image classification model

**Figure 5**
Example of the result of SMILES enumeration and Chemception preprocessing on original SMILES and an augmentation thereof.




(Simonyan and Zisserman 2015). The extracted features from the VGG16 model are in vector shape, so a fully-connected neural network is used for classification using these features. The full model architecture of both approaches can be seen in figure 7 and figure 6.

**Figure 6**
Pre-trained VGG16 feature extraction with dense layers and SoftMax activation attached for prediction.

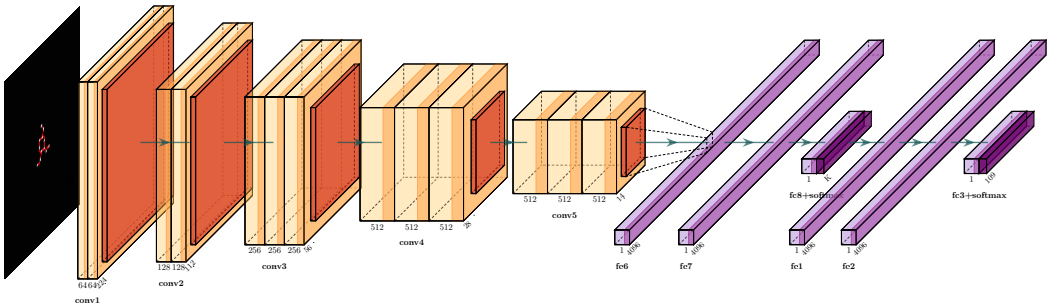
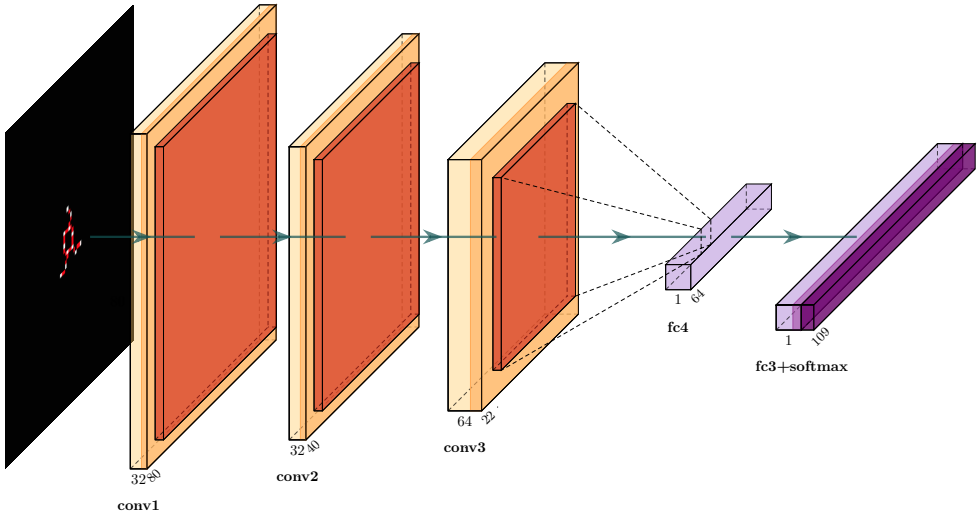


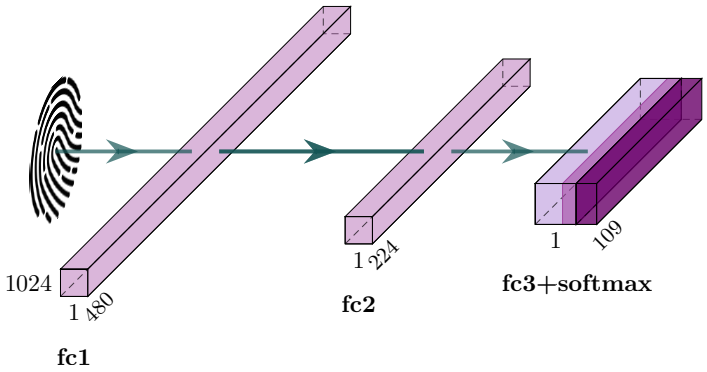
**Figure 7**
Convolutional neural network used to predict on images with chemception preprocessing applied.




**3.3.2 Extended-Connectivity Fingerprints.** Extended-connectivity fingerprints (ECFPs) are topological fingerprints for molecular characterization in QSAR modeling (Rogers and Hahn 2010). Molecular fingerprints are feature representations as bit vectors of a configurable size. For this research, a circular type of ECFPs, Morgan fingerprints, were constructed using the RDKit chem-informatics python package (Landrum 2013). ECFPs are generally used for nearest-neighbors approaches using similarity metrics of multiple fingerprints, but the vectors are also usable as input to a neural network. Two hyper-parameters were tuned in fingerprint creation: number of bits and radius size. After 10-fold cross-validation with values between 512 and 10240 for the number of bits and values between 1 and 10 for radius size, the optimal combination of hyperparameters was found to be 1024 and 2. The resulting bit vectors are highly sparse, with an average of 23.6 non-zero values per 1024 bits.

The fingerprint vectors are used in a fully connected neural network with two layers and dropout regularization in between layers. Figure 8 depicts the model architecture of the model utilizing fingerprint vectors as input.

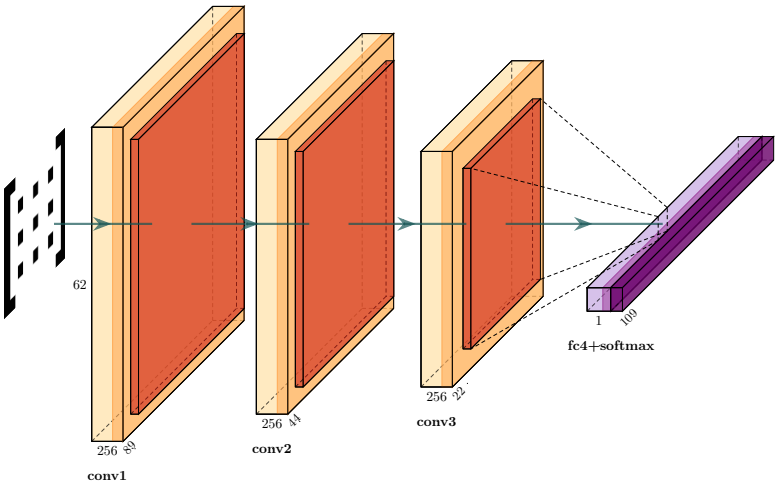
**Figure 8**
Fully connected neural network with softmax classification based on extended-connectivity fingerprints as input.




**3.3.3 Atom and bond tensors.** This research's third input representation is a transformation of SMILES to tensors representing their atom and bond features. This input representation is based on an application of convolutional networks on graphs to learn molecular fingerprints (Duvenaud et al. 2015). These tensors are built using the ChemML Python library (Haghighatlari et al. 2020), and the resulting tensors are of shape (4316, 89, 62) and (4316, 5, 6) for atom and bond features, respectively. Two different input representations are derived from these tensors. The first is a concatenation of each flattened feature set; the second is the non-flattened atom feature set. After 10-fold cross-validation, the best performing representation is found to be the 89 by 62 matrix of atom features.

The atom matrices contain spatial relations, allowing superior classification performance of the atom matrices compared to the flattened atom and bond vectors. Because of these spatial relations, convolutional neural networks are most suitable for prediction with this input representation. The model architecture using atom matrices is drawn out in figure 9.

**Figure 9**
Convolutional neural network with softmax classification based on atom matrix as input.

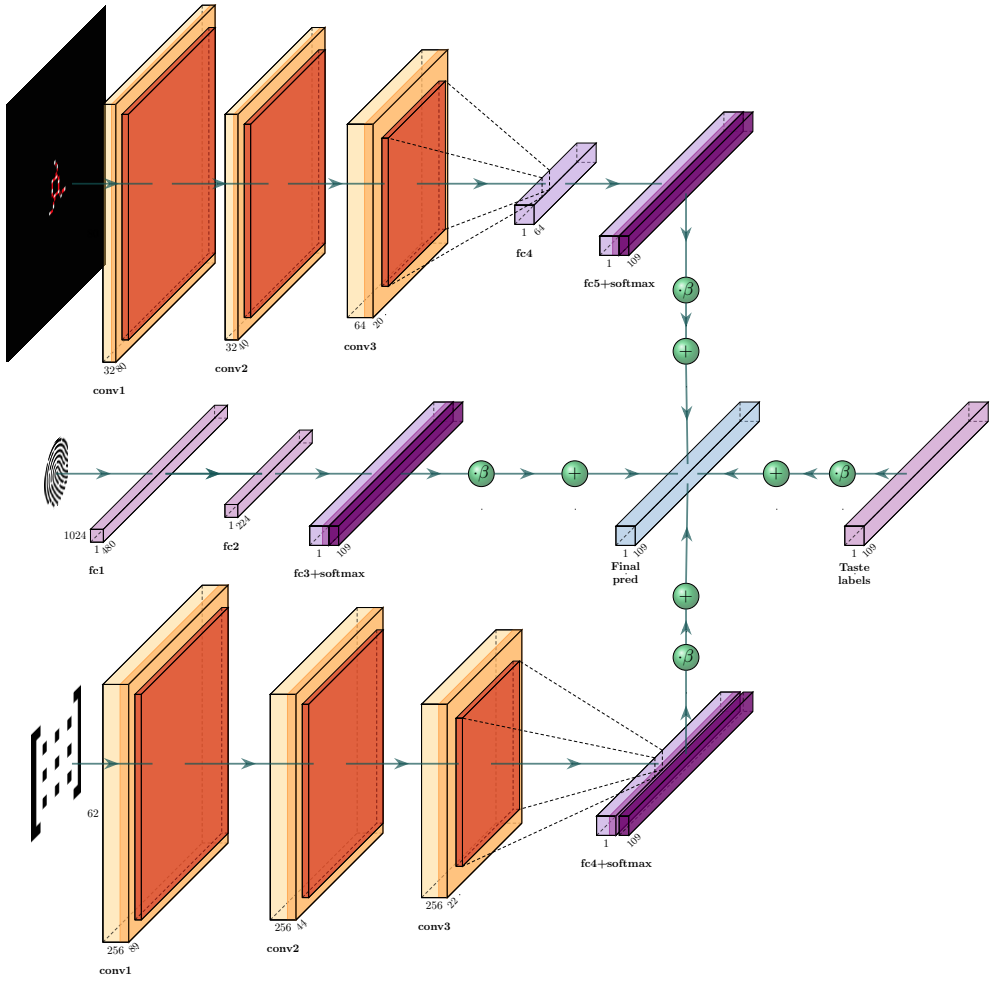



**3.3.4 Conjoining multiple input representations.** Each input representation has its advantages and drawbacks, and the concatenation of multiple models, or even multiple separately trained replicas of the same model, can often lead to significant improvements in prediction performance (Li, Wang, and Ding 2018). In this research, two approaches were explored to use multiple input representations for a final prediction. The first method was to concatenate the three previously mentioned models, up to their penultimate layers, into one model. Effectively, this involved removing the softmax activations from each separate model and attaching the model using Tensorflow concat layers. On top of this concatenation, two densely connected layers were attached. The resulting model could be trained conjointly using all three input representations. The problem with this model was the number of trainable parameters. The model was so complicated that the close to 70 million parameters would lead to overfitting on the training set due to the limited data size.

The second approach involved keeping the softmax activations on each model, training the models separately, and combining their activations into a final prediction. To scale each model's activations, a $\beta$ hyperparameter was introduced for every separate model, with which its activations were multiplied. Training each model separately meant better computational efficiency and less overfitting due to smaller amounts of trainable parameters in each model.

The resulting activation of each model is a vector of softmax probabilities of size 109. As the supplementary data in the form of taste labels is in the shape of a binary vector of size 109, these activations and labels are suitable for addition. Before this addition, another $\beta$ hyperparameter is introduced to scale the taste labels and control for its prevalence in the final prediction. The resulting ensemble of models, including the addition of taste labels and scaling by the $\beta$ hyperparameters is depicted in figure 10.

**Figure 10**
Full ensemble of three models and addition of taste labels to create a final prediction using all available input representations.




### 3.4 Hyperparameter optimization

The hyperparameters in the model architecture were all optimized using a randomized search optimizing for cross-entropy validation loss with the Keras Tuner Python library (O'Malley et al. 2019).

Binary cross-entropy: $-\sum_{i=1}^{N}(y_i \log(p) + (1-y_i)\log(1-p))$
*Where $y$ = binary indicator (correct or incorrect) and $p$ = predicted probability*

Since there are more than two possible labels, the loss is calculated separately for each class label per observation and the result is summed as follows:

Categorical cross-entropy: $-\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$
*Where $M$ = number of classes, $y$ = binary indicator (correct or incorrect classification $c$ for observation $o$) and $p$ = predicted probability of class $c$ given observation $o$*

As an observation can have multiple labels, the one-hot encoded true label vector is normalized by dividing all numbers by the total number of labels for an observation. This means $y$ in the categorical cross-entropy function becomes $\frac{1}{n}$ *where $n$ = number of labels*

Since dropout regularization varies the learning rate indirectly, using a fixed number of epochs would put high values of dropout regularization at a disadvantage. Therefore, models in the randomized search were trained with an early stopping callback meaning they train until validation loss reaches a minimum.

All $\beta$ hyperparameters of the ensemble were optimized using a grid search approach to determine the best possible $\beta$ values using the validation set of 10 separate folds.

### 3.5 Evaluation

The evaluation in this research consists of two sections: overlap between olfactory and taste labels, benchmarking against state of the art QSOR models and external validation. The evaluation metrics will be different per purpose, and will be further explained in the following sections.

**3.5.1 Overlap evaluation.** In order to gauge the overlap between the acquired taste labels and the ground truth olfactory labels in the dataset, a selection of evaluation metrics will be used. The problem at hand is multilabel with only a few labels per molecule out of the possible 109. Therefore, a metric of accuracy is not relevant, as the abundance of negative labels will influence the accuracy metric. Instead, the classification metrics Precision, Recall and F1-score will be used:

$Recall = \frac{TP}{TP+FN}$ $Precision = \frac{TP}{TP+FP}$ $F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$
*Where $TP$ = true positives, $FP$ = false positives and $FN$ = false negatives.*

**3.5.2 Benchmarking.** The right evaluation metric is dependent upon the application of a model. As this research is not centered around a specific use case, metric selection can not be based on this. Instead, metrics will be used that allow for benchmarking against the current state of the art models. The current state of the art in quantitative structure-odor relationship modeling in published papers by Sanchez-Lengeling et al. in 2019 uses a highly similar dataset, with the same label distribution and just slightly more training data. Therefore, the approach is similar enough to benchmark the results of this model against. The aforementioned paper reports f1-scores for model evaluation. The F1-score is a score that considers both the recall and precision scores and combines them into a single metric as follows:

$F1 = \frac{2*Precision*Recall}{Precision+Recall} = \frac{2*TP}{2*TP+FP+FN}$
*Where $TP$ = true positives, $FP$ = false positives and $FN$ = false negatives.*

**3.5.3 External validation.** The same dataset used in this research is used by all participants in the 'learning to smell' challenge. This opens up the opportunity to externally validate the results of this research against other ongoing participants' yet unpublished results. The challenge uses top-5 Tanimoto Similarity Score (TSS) for evaluation. The TSS is based on the amount of overlap between the predictions and ground truth labels as follows:

$$TSS(U, V) = \frac{|U \cap V|}{|U \cup V|}$$

*Where $U$ = predictions, $V$ = ground truth labels, $\cap$ = intersection and $\cup$ = union*

Final reported metrics with their respective confidence boundaries were established using a bootstrapping with replacement approach with 100 repetitions. For every repetition, a random single sample was sampled from the dataset $n$ times, where $n$ is the number of total samples in the dataset. This results in a sample of size $n$ that includes a random selection, which can include multiple values of the same sample. This sample is used for training, while the set of out of bag samples is used for validation. 100 repetitions were found to be a right balance between minimizing estimation error and computation time when models were trained on a Tesla P-100 graphical processing unit.

## 4. Results

In this section, the most essential model evaluation results will be presented and benchmarked against the current state of the art in quantitative structure-odor relationship modeling.

### 4.1 Data augmentation and transfer learning

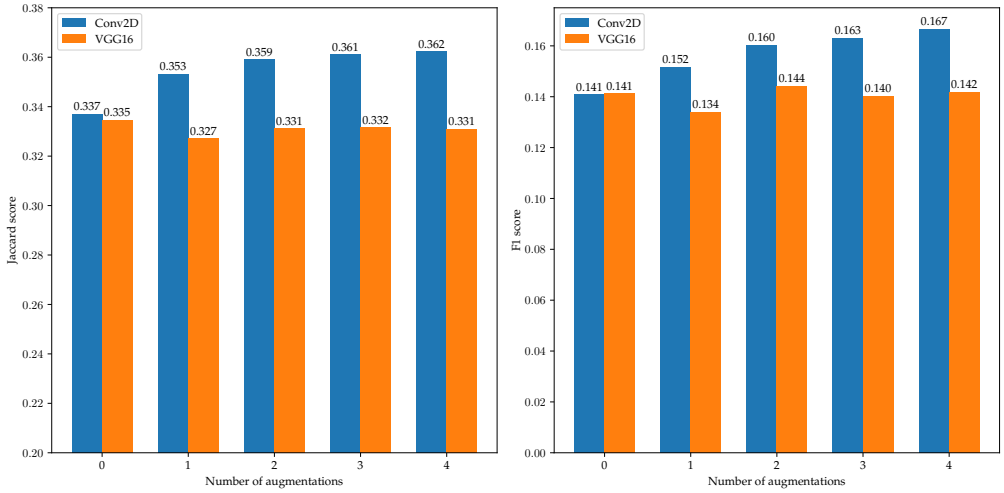
Since the different SMILES enumerations essentially represent the same molecule, the fingerprint vector and atom matrix are identical for all numerations, and data augmentation therefore only entails data duplication. For this reason, the data augmentation approach only affects the models using chemception images as input.

As seen in figure 11, the VGG16 feature extraction approach performs similarly to the approach that learns directly from chemception images as input without any data augmentation. However, when using data augmentation, the model directly using chemception images improves performance with every augmentation added, up to around three augmentations, when additional results are minimal. The VGG16 model does not benefit from data augmentation, as similar features are seemingly extracted from the inputs, and the input diversity is therefore discarded.

### 4.2 Taste labels

The taste labels acquired by web-scraping the Pubchem database as supplementary data show great overlap with the true odor labels in the dataset. The full overlap of both methods of label generation is depicted in table 1. Because of the stricter algorithm, the average number of positive labels for the NLP method (3) is far smaller than the average number of positive labels for the Naive method (47). This has consequences for the precision and recall scores of each method: the Naive method scores best on recall, while the NLP method optimizes for precision. Since the problem is multilabel with 109 labels, there are 109 confusion matrices for each method. The full confusion matrices

**Figure 11**
Comparison of two models using varying numbers of augmentations on the chemception images (results from 10-fold cross-validation).




per label are depicted in Appendix D: Confusion matrices per label (Naive method) and Appendix D: Confusion matrices per label (NLP method). These confusion matrices per label show that there is great variability in the overlap between odor and taste labels; some labels exactly overlap, while others show no overlap at all.

---

**Table 1**
Micro-averaged confusion matrices and classification report including Precision, Recall, and F1. Values are derived by treating the automatically generated taste labels as odor predictions.

**Naive method**

| Taste labels | p | n | total |
|---|---|---|---|
| p′ | TP 13.7 | FP 33.5 | 47 |
| n′ | FN 103.4 | TN 4165.3 | 4269 |
| total | 117 | 4199 | |

**NLP method**

| Taste labels | p | n | total |
|---|---|---|---|
| p′ | TP 1.5 | FP 1.5 | 3 |
| n′ | FN 115.6 | TN 4197.3 | 4313 |
| total | 117 | 4199 | |

| Method | Precision-score | Recall-score | F1-score |
|---|---|---|---|
| Naive | 0.291 | 0.117 | 0.167 |
| NLP | 0.500 | 0.013 | 0.026 |

**4.2.1 Supplementary dataset.** The supplementary dataset created by web-scraping 5,000,000 entries of the Pubchem database and applying the Naive and NLP methods was used to test model generalization. The Naive method yielded a total of 5,862 molecules with at least one positive label and a valid SMILES representation. The NLP method, with its stricter algorithm, yielded only 23 molecules, and was therefore unfit for neural network training. Model training on the 5,862 SMILES representations from the Naive method showed no convergence. The model fit on the training set of 80% of the data could not generalize to predict on the validation set of 20% of the data. More

information on the model training can be found in Appendix D: Taste dataset training results.

### 4.3 Bootstrapping

Since the 10-fold validations were used for hyperparameter optimization, the metrics from these folds can not be reported as final metrics, as that would entail optimizing and testing on the same subset of data. Therefore, to accurately compare model performance with minimal bias, a bootstrapping approach was applied with 100 repetitions. The results of this bootstrapping evaluation can be found in figure 12 and table 2, which show boxplots and mean values of the evaluation metrics per model.

**Figure 12**
Standard boxplots of evaluation metrics from 100 repetitions of bootstrapping with boxes representing interquartile range, orange lines the medians, whiskers the full data ranges excluding outliers and circles representing outliers. The models represented are: Chemception images (1); Fingerprints (2); Atom matrices (3); Ensemble (4); Ensemble + taste labels (Naive) (5). A table of the same data can be seen in table 2.

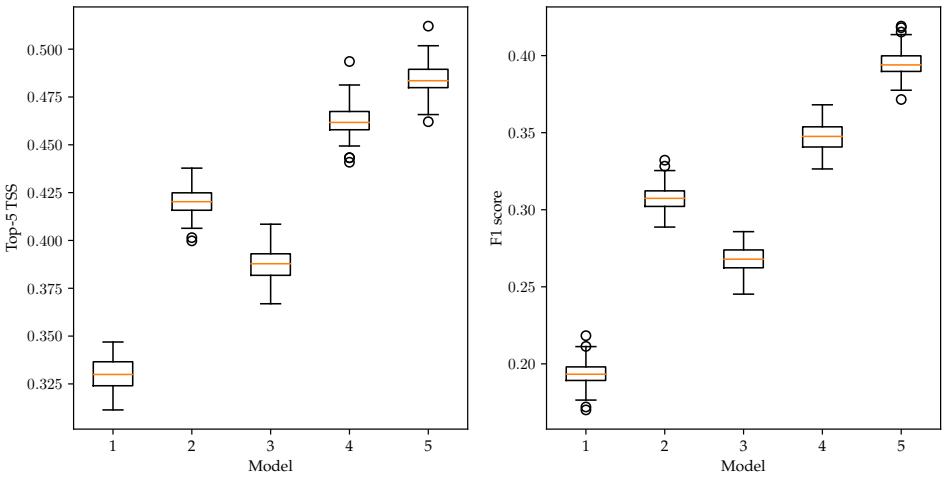



**Table 2**
Evaluation metrics per model from 100 repetitions of bootstrapping, including 95% confidence intervals. Boxplot representations of the same data can be seen in figure 12.

| Model | Top-5 TSS | F1-score |
|---|---|---|
| Chemception images (1) | 0.330 [0.328-0.332] | 0.193 [0.192-0.195] |
| Fingerprints (2) | 0.420 [0.419-0.422] | 0.308 [0.306-0.309] |
| Atom matrices (3) | 0.387 [0.386-0.389] | 0.268 [0.267-0.269] |
| Ensemble (4) | 0.463 [0.461-0.464] | 0.348 [0.346-0.349] |
| Ensemble + taste labels (Naive) (5) | **0.484 [0.483-0.486]** | **0.395 [0.393-0.397]** |
| GNN (Sanchez-Lengeling et al. 2019) | n.a. | 0.360 [0.337-0.372] |

From these metrics, it is clear that none of the separate models can perform prediction at the current state of the art level (Sanchez-Lengeling et al. 2019). Performance of the ensemble model without addition of the taste labels comes closer to this state of the art. While the mean F1-score is slightly sub-par to GNN, the evaluation metrics are indistinguishable when taking into account confidence boundaries of the bootstrapping distribution.

Once the supplementary taste labels are added to the model, the model outperforms GNN in F1-score evaluation, even when considering the confidence boundaries of the distribution.

### 4.4 External validation

Operating on the test set is not allowed for the 'learning to smell' challenge, and the inability to access the internet during model evaluation makes it technically impossible to use the taste labels for prediction in this challenge. Therefore, the model ensemble not including taste labels is used for external validation.

The top-5 Tanimoto similarity score for this model's prediction on the previously unseen test set for which the labels are unknown is 0.454 (#4/670 participants at the moment of writing). This is very close to the internally validated score of 0.463, which validates the ability to generalize to completely unseen data.

The top participant (#1/670 at the moment of writing) scores 0.514. The model architecture and validity of this prediction is unknown, as the challenge is still ongoing at the moment of writing.

## 5. Discussion and conclusions

Three subquestions were formulated to help answer the main research question. The findings of the experiments will be analyzed in this chapter to answer these subquestions and aid in the discussion of the main research question.

### 5.1 SQ1

*SQ1: Which standard molecular input representations yield the strongest predictive value for olfaction modeling?*

From the three standard molecular input representations in this research, the chemception image preprocessing approach yielded the worst predictive value for olfaction modeling. This result is contrary to expectations, as chemception is known to perform at a similar level to more common input representations in tasks such as solvation and toxicity prediction (Goh et al. 2017). Interestingly, VGG16 feature extraction did not improve the prediction performance on chemception images. This is unexpected, as transfer learning can often reap great benefits in predictive power when datasets are small. A reason for the poor performance of VGG16 feature extraction could be that the VGG16 model was pre-trained on the ImageNet dataset, which contains mostly photography (Simonyan and Zisserman 2015), and is therefore very different from the chemception molecular drawings.

From the experiments in this research, the extended-connectivity fingerprints show the best predictive value for olfaction modeling, and the respective model is also the most computationally efficient. ECFPs have been the state of the art in QSAR for many years when standard input representations are concerned (Rogers and Hahn 2010), so the excellent predictive power and computational efficiency is unsurprising.

The atom matrices used with a convolutional neural network perform subpar to ECFPs, but only by a slight margin. Although these atom matrices have outperformed molecular fingerprints in different tasks (Duvenaud et al. 2015), these matrices are not optimized for smell prediction, which can be an explanation for the decline in performance compared to ECFPs.

### 5.2 SQ2

*SQ2: Can the varying benefits of multiple molecular input representations be utilized conjointly to improve smell prediction performance to state of the art level?*

The input representations each have their benefits and drawbacks, which is why the ensemble of models far outperformed the best single model as it can use all these benefits conjointly.

While the chemception images performed the worst as a single model, the representation is so different from the two other representations since they are a direct representation instead of an embedding. This could be the reason that it adds great predictive performance to the ensemble model.

The ensemble of models increased the mean performance based on the evaluation metrics by 10% and 12% for Tanimoto similarity and F1 score respectively. An increase in performance was hypothesized, but the extent is greater than expected, as common performance increases from model ensembles are often less strong (Li, Wang, and Ding 2018). A possible explanation for better performance increase compared to the literature is that many ensembles are usually bagging ensembles, which entail multiple training runs of the same model, or a slightly different model architecture. In this case, the entire modeling pipeline of the three models is different from preprocessing to model architecture. This great diversity might have led to a more significant increase in prediction performance.

### 5.3 SQ3

*SQ3: Can the overlap between labels used to describe molecules' taste and smell be exploited to improve olfaction prediction performance?*

There is significant overlap between the labels used to describe smells and taste, as seen in the confusion matrices of the taste labels in the results section. This overlap was hypothesized a priori, as the mechanisms for smell and taste work so similarly (Buck 2006). Despite the false positive problem of the naively acquired taste labels, incorporating the labels into the ensemble model as a statistical prior led to a significant increase in model performance on Tanimoto similarity as well as F1 score, outperforming the state of the art graph neural network approach which is highly dependent on domain knowledge on F1 score. The overlap between labels used to describe molecoles' taste and smell can therefore be exploited to improve olfaction prediction performance.

The overlap is proven to be useful in prediction performance improvements by adding the taste vector to the smell prediction vector as described in section 3.3.4. However, the results of training a model on the new data show that the data holds too little predictive value to train a neural network model on. This is likely due to the false positive problem of the Naive method, and the false negative problem of the NLP method: the Naive method produces too many incorrect positive labels, which contaminate the validity of the dataset. The NLP method produces so little positive labels because of its strict algorithm, that the dataset is too small for neural network training.

### 5.4 Main research question

*RQ: Can modern deep learning algorithms using standard molecular input representations perform molecular smell prediction on par with state of the art quantitative structure-odor relationship models?*

To answer the main research question with an unequivocal yes, a model would need to be developed that consists purely of modeling based on the chemical structure that outperforms the current state of the art GNN approach (Sanchez-Lengeling et al. 2019) with consideration of confidence boundaries.

The three standard input representations used in this research are all unable to perform molecular smell prediction on par with state of the art quantitative structure-odor relationship models on their own, as their F1 scores are all lower than GNN, even when considering confidence boundaries.

Concerning the ensemble of models, not including the supplementary taste labels, the mean performance is slightly sub-par to GNN. Although the confidence boundaries overlap slightly, it cannot be stated definitively that this ensemble of models performs olfaction prediction at the same level as GNN.

The ensemble of models incorporating the taste labels outperforms GNN on F1 score validation, even when taking into account confidence boundaries of the bootstrapping repetitions. However, this final prediction is partially based on the automated collection of supplementary data in the form of taste labels and is therefore not of a fully chemical-structure-modeling nature. While the approach pipeline can be fully automated to predict on completely unseen data, making it feasible for practical applications, the goal of benchmarking against the current state of the art quantitative structure-odor relationship models is to improve upon existing molecular data representations and thereby contribute to this scientific domain.

### 5.5 Future work

The models explored in this research prove the ability of modeling for smell prediction with minimal application of domain knowledge is possible using standard molecular representations created with cheminformatics software. While the results are novel considering scientific progress, the reported evaluation metrics reap little societal implications, as the levels of performance are unlikely to be at a level required for any practical application yet. The findings of this study open up opportunities for new research in many different aspects.

The problem in this study is one of multiclass classification with 109 total labels and only 4.316 total data samples in the train/validation set. This ratio between the number of labels and data samples is significant and is likely one of the main reasons for the poor performance compared to many other deep learning applications. To control this factor, both sides of the ratio could be influenced in future research.

As the labels are highly related and co-occurrence is great between sets of related labels, it would be interesting to explore the same problem with the same dataset while reducing the maximum number of labels by grouping more specific labels into less specific labels. Prediction performance is likely to rise from this new label set, and the ability could arise for domain applications that require less specificity to predict more accurately.

The model ensemble using standard molecular input representations might prove useful in different tasks in QSAR. Future researchers could approach a supervised

learning task using molecular SMILES as primary input by using the pipeline presented in this thesis to find out if the model can generalize to similar, but different tasks.

## References


Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, and Google Brain. Tensorflow: A system for large-scale machine learning tensorflow: A system for large-scale machine learning.

Bjerrum, Esben Jannik. 2017. Smiles enumeration as data augmentation for neural network modeling of molecules. *arXiv*.

Bloice, Marcus D., Christof Stocker, and Andreas Holzinger. 2017. Augmentor: An image augmentation library for machine learning. *arXiv*.

Boelens, Mans. Database of perfumery materials & performance.

Buck, L. 2006. Smell and taste : The chemical senses.

Chowdhary, K. R. 2020. Natural language processing. *Fundamentals of Artificial Intelligence*, pages 603–649.

Dash, Tirtharaj, Ashwin Srinivasan, and Lovekesh Vig. 2020. Incorporating symbolic domain knowledge into graph neural networks.

Dozat, Timothy and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing.

Duvenaud, David, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints.

Gardner, Julian W. and Philip N. Bartlett. 1994. A brief history of electronic noses. *Sensors and Actuators: B. Chemical*, 18:210–211.

Genva, Manon, Tierry Kenne Kemene, Magali Deleu, Laurence Lins, and Marie Laure Fauconnier. 2019. Is it possible to predict the odor of a molecule on the basis of its structure? *International Journal of Molecular Sciences*, 20.

Goh, Garrett B., Charles Siegel, Abhinav Vishnu, Nathan O. Hodas, and Nathan Baker. 2017. Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv*.

Haghighatlari, Mojtaba, Gaurav Vishwakarma, Doaa Altarawy, Ramachandran Subramanian, Bhargava U. Kota, Aditya Sonpal, Srirangaraj Setlur, and Johannes Hachmann. 2020. chemml : A machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data. *WIREs Computational Molecular Science*, 10.

Kell, Alexander J.E., Daniel L.K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. 2018. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98:630–644.e16.

Kheradpisheh, Saeed Reza, Masoud Ghodrati, Mohammad Ganjtabesh, and Timothée Masquelier. 2016. Deep networks can resemble human feed-forward vision in invariant object recognition. *Scientific Reports*, 6.

Kim, Sunghwan, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A. Shoemaker, Paul A. Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E. Bolton. 2019. Pubchem 2019 update: Improved access to chemical data. *Nucleic Acids Research*, 47:D1102–D1109.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 2:1–127.

Landrum, Gregoy. 2013. Getting started with the rdkit in python — the rdkit 2020.09.1 documentation.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521:436–444.

Li, Hui, Xuesong Wang, and Shifei Ding. 2018. Research and development of neural network ensembles: a survey. *Artificial Intelligence Review*, 49:455–479.

Mohanty, Sharada.

O'Malley, Tom, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, et al. 2019. Keras Tuner. https://github.com/keras-team/keras-tuner.

Rogers, David and Mathew Hahn. 2010. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50:742–754.

Sainath, Tara N., Oriol Vinyals, Andrew Senior, and Hasim Sak. 2015. Convolutional, long short-term memory, fully connected deep neural networks. volume 2015-August, pages 4580–4584, Institute of Electrical and Electronics Engineers Inc.

Sanchez-Lengeling, Benjamin, Jennifer N. Wei, Brian K. Lee, Richard C. Gerkin, Alán Aspuru-Guzik, and Alexander B. Wiltschko. 2019. Machine learning for scent: Learning generalizable perceptual representations of small molecules. *arXiv*.

Simonyan, Karen and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations, ICLR.

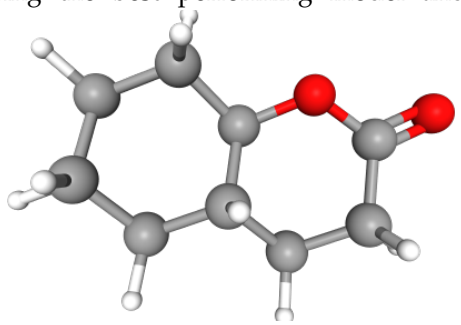
Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. pages 4278–4284, AAAI press.

Wu, Zhenqin, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. 2018. Moleculenet: A benchmark for molecular machine learning. *Chemical Science*, 9:513–530.

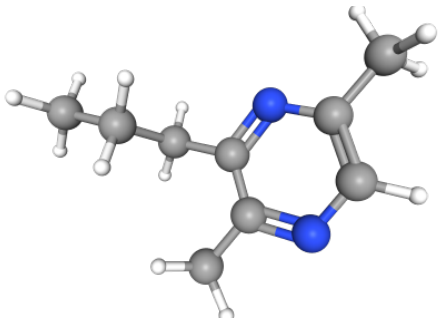
Zhang, Zhisong, Hai Zhao, and Lianhui Qin. 2016. Probabilistic graph-based dependency parsing with convolutional neural network. pages 1382–1392.

**Appendix A: Prediction examples**

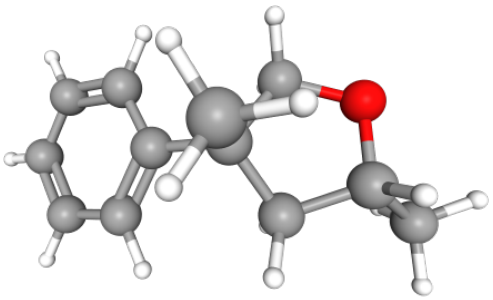
This section serves to give insight in some examples of how different methods come to different predictions in this research. For three molecules, the 3D structure is shown along with its respective SMILES respresentation and predictions using the best performing model and both methods of taste label acquisition.



| | |
|---|---|
| SMILES | O=C1CCC2C(O1)CCCC2 |
| True labels | Fruity, vanilla, coconut, sharp |
| Ensemble predictions | Fruity, sweet, vanilla |
| Naive method taste | Sweet, spicy |
| NLP method taste | Sweet |



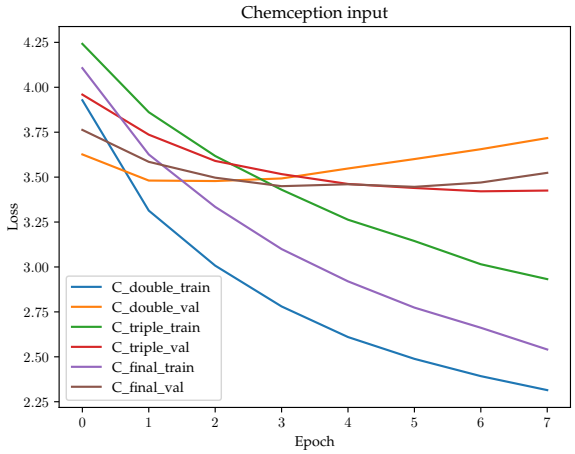
| | |
|---|---|
| SMILES | CCCc1c(C)ncc(C)n1 |
| True labels | Nut |
| Ensemble predictions | Nut, hazelnut |
| Naive method taste | Hazelnut |
| NLP method taste | Hazelnut |

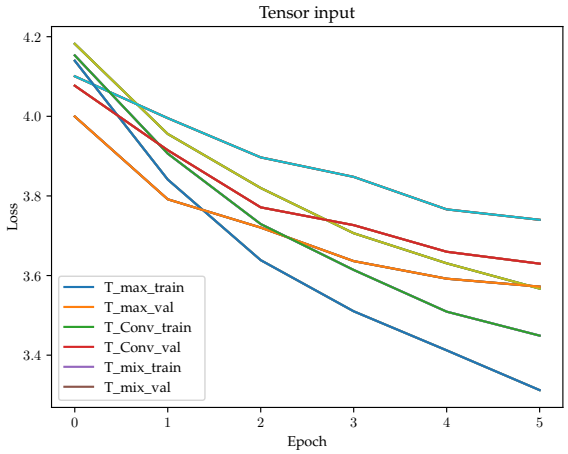


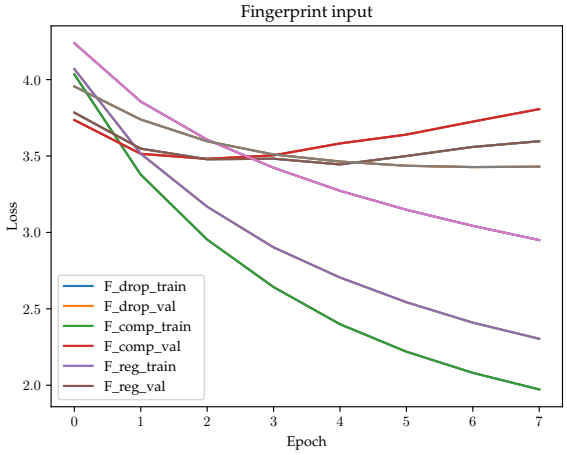
| | |
|---|---|
| SMILES | CC1CC(C)(CO1)c1ccccc1 |
| True labels | Green, fruity, grapefruit, camphor, woody |
| Ensemble predictions | Grapefruit, camphor, woody |
| Naive method taste | Grapefruit, grass, herbal |
| NLP method taste | Grapefruit |

## Appendix B: Model comparison

**Figure 1**
Results of model comparison for each type of input.



**Appendix C: Confusion matrices per label (Naive method)**

| | True negative | False positive | False negative | True positive |
|---|---|---|---|---|
| alcoholic | 4233 | 61 | 20 | 2 |
| aldehydic | 4200 | 7 | 104 | 5 |
| alliaceous | 4222 | 0 | 87 | 7 |
| almond | 4236 | 16 | 49 | 15 |
| ambergris | 4292 | 2 | 20 | 2 |
| ambery | 4241 | 0 | 75 | 0 |
| ambrette | 4302 | 6 | 6 | 2 |
| ammoniac | 4308 | 0 | 8 | 0 |
| animalic | 4210 | 0 | 106 | 0 |
| anisic | 4268 | 2 | 44 | 2 |
| apple | 4149 | 35 | 111 | 21 |
| balsamic | 4026 | 20 | 245 | 25 |
| banana | 4257 | 12 | 37 | 10 |
| berry | 4141 | 22 | 137 | 16 |
| blackcurrant | 4257 | 10 | 44 | 5 |
| blueberry | 4301 | 9 | 4 | 2 |
| body | 4051 | 227 | 34 | 4 |
| bread | 4286 | 13 | 13 | 4 |
| burnt | 4164 | 22 | 118 | 12 |
| butter | 4244 | 22 | 42 | 8 |
| cacao | 4239 | 2 | 75 | 0 |
| camphor | 4128 | 20 | 148 | 20 |
| caramellic | 4193 | 3 | 115 | 5 |
| cedar | 4275 | 8 | 31 | 2 |
| cheese | 4236 | 26 | 39 | 15 |
| chemical | 4201 | 0 | 115 | 0 |
| cherry | 4269 | 14 | 23 | 10 |
| cinnamon | 4246 | 22 | 43 | 5 |
| citrus | 4036 | 67 | 187 | 26 |
| clean | 4163 | 82 | 62 | 9 |
| clove | 4278 | 6 | 29 | 3 |
| coconut | 4256 | 15 | 39 | 6 |
| coffee | 4183 | 70 | 46 | 17 |
| cognac | 4298 | 6 | 11 | 1 |
| coniferous | 4264 | 0 | 52 | 0 |
| cooked | 4247 | 18 | 31 | 20 |
| cooling | 4238 | 15 | 58 | 5 |
| cucumber | 4291 | 6 | 14 | 5 |
| dairy | 4187 | 49 | 72 | 8 |
| dry | 4007 | 198 | 94 | 17 |
| earthy | 4039 | 43 | 210 | 24 |
| ester | 4275 | 0 | 41 | 0 |
| ethereal | 4084 | 16 | 202 | 14 |
| fatty | 3877 | 268 | 132 | 39 |
| fennel | 4305 | 4 | 6 | 1 |

| | | | | |
|---|---|---|---|---|
| fermented | 4239 | 12 | 58 | 7 |
| floral | 3581 | 103 | 537 | 95 |
| fresh | 3623 | 189 | 466 | 38 |
| fruity | 3242 | 182 | 749 | 143 |
| geranium | 4259 | 14 | 31 | 12 |
| gourmand | 4278 | 0 | 38 | 0 |
| grape | 4251 | 24 | 35 | 6 |
| grapefruit | 4262 | 9 | 37 | 8 |
| grass | 4227 | 9 | 79 | 1 |
| green | 3593 | 167 | 439 | 117 |
| herbal | 3693 | 59 | 515 | 49 |
| honey | 4209 | 22 | 64 | 21 |
| hyacinth | 4261 | 10 | 39 | 6 |
| jasmin | 4236 | 6 | 67 | 7 |
| lactonic | 4277 | 3 | 36 | 0 |
| leaf | 4198 | 23 | 84 | 11 |
| leather | 4238 | 29 | 47 | 2 |
| lemon | 4198 | 21 | 91 | 6 |
| lily | 4198 | 7 | 102 | 9 |
| liquor | 4258 | 3 | 55 | 0 |
| meat | 4164 | 22 | 117 | 13 |
| medicinal | 4241 | 15 | 54 | 6 |
| melon | 4245 | 19 | 40 | 12 |
| metallic | 4234 | 14 | 60 | 8 |
| mint | 4125 | 19 | 158 | 14 |
| mushroom | 4257 | 13 | 36 | 10 |
| musk | 4170 | 5 | 131 | 10 |
| musty | 4168 | 34 | 95 | 19 |
| nut | 4129 | 16 | 154 | 17 |
| odorless | 4254 | 5 | 53 | 4 |
| oily | 3997 | 138 | 169 | 12 |
| orange | 4198 | 48 | 60 | 10 |
| overripe | 4305 | 1 | 10 | 0 |
| pear | 4248 | 11 | 45 | 12 |
| pepper | 4236 | 38 | 40 | 2 |
| phenolic | 4176 | 17 | 97 | 26 |
| plastic | 4204 | 94 | 13 | 5 |
| plum | 4271 | 11 | 30 | 4 |
| powdery | 4172 | 23 | 104 | 17 |
| pungent | 4176 | 39 | 93 | 8 |
| rancid | 4265 | 7 | 42 | 2 |
| resinous | 3945 | 1 | 369 | 1 |
| ripe | 4286 | 10 | 17 | 3 |
| roasted | 4180 | 64 | 45 | 27 |
| rose | 4004 | 54 | 223 | 35 |
| seafood | 4284 | 1 | 30 | 1 |
| sharp | 4277 | 14 | 20 | 5 |
| smoky | 4265 | 13 | 35 | 3 |
| sour | 4243 | 0 | 73 | 0 |

| | | | | |
|---|---|---|---|---|
| spicy | 3964 | 50 | 263 | 39 |
| sulfuric | 4137 | 22 | 156 | 1 |
| sweet | 3586 | 279 | 351 | 100 |
| syrup | 4265 | 17 | 31 | 3 |
| terpenic | 4269 | 0 | 45 | 2 |
| tobacco | 4212 | 45 | 52 | 7 |
| tropicalfruit | 4144 | 0 | 172 | 0 |
| vanilla | 4201 | 14 | 82 | 19 |
| vegetable | 4159 | 22 | 105 | 30 |
| violetflower | 4234 | 0 | 82 | 0 |
| watery | 4283 | 1 | 32 | 0 |
| waxy | 4133 | 35 | 119 | 29 |
| whiteflower | 4279 | 0 | 37 | 0 |
| wine | 4247 | 20 | 43 | 6 |
| woody | 3662 | 66 | 544 | 44 |

**Appendix D: Confusion matrices per label (NLP method)**

| | True negative | False positive | False negative | True positive |
|---|---|---|---|---|
| alcoholic | 4293 | 1 | 21 | 1 |
| aldehydic | 4205 | 2 | 107 | 2 |
| alliaceous | 4222 | 0 | 93 | 1 |
| almond | 4252 | 0 | 58 | 6 |
| ambergris | 4294 | 0 | 21 | 1 |
| ambery | 4241 | 0 | 75 | 0 |
| ambrette | 4308 | 0 | 8 | 0 |
| ammoniac | 4308 | 0 | 8 | 0 |
| animalic | 4210 | 0 | 106 | 0 |
| anisic | 4270 | 0 | 46 | 0 |
| apple | 4179 | 5 | 128 | 4 |
| balsamic | 4045 | 1 | 268 | 2 |
| banana | 4269 | 0 | 44 | 3 |
| berry | 4160 | 3 | 148 | 5 |
| blackcurrant | 4266 | 1 | 46 | 3 |
| blueberry | 4310 | 0 | 6 | 0 |
| body | 4278 | 0 | 38 | 0 |
| bread | 4299 | 0 | 16 | 1 |
| burnt | 4181 | 5 | 127 | 3 |
| butter | 4264 | 2 | 49 | 1 |
| cacao | 4241 | 0 | 75 | 0 |
| camphor | 4148 | 0 | 167 | 1 |
| caramellic | 4196 | 0 | 118 | 2 |
| cedar | 4283 | 0 | 33 | 0 |
| cheese | 4261 | 1 | 51 | 3 |
| chemical | 4201 | 0 | 115 | 0 |
| cherry | 4282 | 1 | 33 | 0 |
| cinnamon | 4268 | 0 | 48 | 0 |
| citrus | 4099 | 4 | 204 | 9 |
| clean | 4243 | 2 | 69 | 2 |
| clove | 4284 | 0 | 32 | 0 |
| coconut | 4271 | 0 | 44 | 1 |
| coffee | 4253 | 0 | 63 | 0 |
| cognac | 4304 | 0 | 12 | 0 |
| coniferous | 4264 | 0 | 52 | 0 |
| cooked | 4264 | 1 | 40 | 11 |
| cooling | 4251 | 2 | 62 | 1 |
| cucumber | 4297 | 0 | 19 | 0 |
| dairy | 4236 | 0 | 78 | 2 |
| dry | 4203 | 2 | 108 | 3 |
| earthy | 4079 | 3 | 234 | 0 |
| ester | 4275 | 0 | 41 | 0 |
| ethereal | 4099 | 1 | 215 | 1 |
| fatty | 4143 | 2 | 166 | 5 |
| fennel | 4309 | 0 | 7 | 0 |

| | | | | |
|---|---|---|---|---|
| fermented | 4251 | 0 | 65 | 0 |
| floral | 3676 | 8 | 622 | 10 |
| fresh | 3811 | 1 | 502 | 2 |
| fruity | 3408 | 16 | 869 | 23 |
| geranium | 4273 | 0 | 42 | 1 |
| gourmand | 4278 | 0 | 38 | 0 |
| grape | 4275 | 0 | 41 | 0 |
| grapefruit | 4271 | 0 | 44 | 1 |
| grass | 4236 | 0 | 79 | 1 |
| green | 3755 | 5 | 553 | 3 |
| herbal | 3751 | 1 | 563 | 1 |
| honey | 4231 | 0 | 83 | 2 |
| hyacinth | 4271 | 0 | 45 | 0 |
| jasmin | 4242 | 0 | 74 | 0 |
| lactonic | 4280 | 0 | 36 | 0 |
| leaf | 4221 | 0 | 95 | 0 |
| leather | 4267 | 0 | 49 | 0 |
| lemon | 4219 | 0 | 97 | 0 |
| lily | 4205 | 0 | 111 | 0 |
| liquor | 4261 | 0 | 55 | 0 |
| meat | 4185 | 1 | 128 | 2 |
| medicinal | 4256 | 0 | 59 | 1 |
| melon | 4264 | 0 | 50 | 2 |
| metallic | 4248 | 0 | 68 | 0 |
| mint | 4144 | 0 | 172 | 0 |
| mushroom | 4270 | 0 | 46 | 0 |
| musk | 4174 | 1 | 139 | 2 |
| musty | 4201 | 1 | 114 | 0 |
| nut | 4145 | 0 | 171 | 0 |
| odorless | 4258 | 1 | 57 | 0 |
| oily | 4134 | 1 | 181 | 0 |
| orange | 4246 | 0 | 70 | 0 |
| overripe | 4306 | 0 | 10 | 0 |
| pear | 4259 | 0 | 55 | 2 |
| pepper | 4274 | 0 | 42 | 0 |
| phenolic | 4193 | 0 | 123 | 0 |
| plastic | 4298 | 0 | 18 | 0 |
| plum | 4281 | 1 | 34 | 0 |
| powdery | 4195 | 0 | 121 | 0 |
| pungent | 4214 | 1 | 101 | 0 |
| rancid | 4272 | 0 | 44 | 0 |
| resinous | 3946 | 0 | 370 | 0 |
| ripe | 4296 | 0 | 20 | 0 |
| roasted | 4244 | 0 | 70 | 2 |
| rose | 4058 | 0 | 258 | 0 |
| seafood | 4285 | 0 | 31 | 0 |
| sharp | 4291 | 0 | 25 | 0 |
| smoky | 4278 | 0 | 38 | 0 |
| sour | 4243 | 0 | 73 | 0 |

| | | | | |
|---|---|---|---|---|
| spicy | 4013 | 1 | 302 | 0 |
| sulfuric | 4159 | 0 | 157 | 0 |
| sweet | 3778 | 87 | 415 | 36 |
| syrup | 4282 | 0 | 34 | 0 |
| terpenic | 4269 | 0 | 47 | 0 |
| tobacco | 4255 | 2 | 59 | 0 |
| tropicalfruit | 4144 | 0 | 172 | 0 |
| vanilla | 4215 | 0 | 99 | 2 |
| vegetable | 4181 | 0 | 135 | 0 |
| violetflower | 4234 | 0 | 82 | 0 |
| watery | 4284 | 0 | 32 | 0 |
| waxy | 4168 | 0 | 148 | 0 |
| whiteflower | 4279 | 0 | 37 | 0 |
| wine | 4267 | 0 | 49 | 0 |
| woody | 3727 | 1 | 587 | 1 |

## Appendix E: Taste dataset training results

**Figure 1**
Results of model training on the supplementary taste dataset. Model used: Fingerprint model