Images of molecules are widely used to predict molecule properties in teaching and chemical research. A trained chemist can easily derive molecule properties by analyzing its structure and evaluate its functional groups. To predict, for example, the water solubility of an organic compound a chemist would intuitively count the number of polar groups, consider the size of the molecule, and estimate the water/molecule interaction by counting the number of H-bond donors and acceptors. Therefore, 2D molecule representations and their directly accessible features should provide enough information to predict the molecule’s structure-dependent properties. To support this thesis, different image-based machine learning approaches as dense neural networks, convolutional neural networks, clustering, data augmentation, and transfer learning are compared and evaluated in this work. The influence of the image size as well as the network size is discussed. Finally, a simple yet effective dense neural network trained on expert preselected, visually accessible features, is presented and its efficiency and comparability to other more complex methods are demonstrated.

1. Introduction

Deep learning in chemical research gains in importance at a rapid pace and the prediction of molecule properties like the free solvation energy\(^1\) or the ionization energy\(^2\) competes with or even outperforms modern ab initio calculations.

The spectrum of possible tasks for machine learning (ML) algorithms seems inexhaustible and can range from the classification of bioactive compounds into active and non-active categories in HIV replication inhibition\(^3\) to regressions tasks where based on the input a specific value, such as the lipophilicity, is predicted.\(^4\)

However, ML algorithms are not limited to the prediction of categories and values. They can generate new compound structures based on input structures or desired interaction targets. Therefore they can support the design of novel molecules in material sciences and medicinal chemistry.\(^5,6\) Moreover, clustering processes divide large molecule datasets into groups of maximum congruence and can be utilized in medicinal chemistry to discover novel lead structures and accelerate their computer aided selection.\(^7\)

For all these tasks molecules must be converted into machine readable inputs, before feeding them to the ML algorithms. Starting from a structural formula the molecule is often converted into a vector to be processable by the most ML algorithms. Those vectors should be unique. Therefore, one vector represents only one compound, invariant to atom numbering and fixed in length. There are several methods available to obtain such inputs and in the following the most popular ones will be discussed.

A simple way to represent a molecular structure is the usage of strings. A string is a series of numbers and letters and often simplified molecular input line entry specification (SMILES) strings are utilized. In these strings each molecule is described by its edge atoms and adjacent bonds and a molecule like n-propanol would be represented by the SMILES-string “CCCO.”\(^8\) Due to the lack of adequately describing chirality, except central chirality, and tautomerism, the IUPAC proposed a different string representation named the International Chemical Identifier (InChI) which is not directly interpretable for a chemist.\(^9\)

Another method of representing a molecular structure are molecular descriptors. Those descriptors can range from simple, from the structure directly accessible, features like the number of atoms, the number of heteroatoms or the molecular weight to complexly calculate or simulate physical properties like the dipole moment and the partial charges.\(^10\) Those derived or calculated properties are then condensed in a defined order into a fixed length vector.

More advanced featurization methods like molecular fingerprints are available and widely used. A molecular fingerprint is a fixed length vector in which each molecule is represented by a sequence of zeros for the absence of a feature and by a sequence of ones for the presence of a feature. Those features can again be directly derived from the structure of a molecule or be automatically obtained by using pre-defined queries of structural motifs. Structure derived fingerprints are called substructure keys fingerprints and were originally developed for substructure and
similarity search in compound databases. Topological fingerprints are a different type of molecular fingerprint and the most common one is the extended-connectivity fingerprint (ECFP) using the Morgan algorithm. To obtain an ECFP each atom of a molecule is assigned with an initial identifier such as the atom type and its connectivity. After the initial assignment of the identifiers, each identifier is updated by including the identifiers of its direct neighbor atoms. By utilizing more update steps and varying the update radius each atom identifier contains more information of its chemical surroundings. A vector of the identifier is created and hashed to a fingerprint which is not directly interpretable on its own.

It is also possible to represent a 3D molecule by a 3D representation of its structure using the coordinates of the atoms, the bond length, and the bond as well as its dihedral-angles. One approach to this type of representation is the Coulomb matrix which describes a molecular structure by its pair-wise Coulomb potentials. This representation method is often used for quantum chemical calculations and can be simplified by the bag of bonds (BOB) representation where only the Coulomb elements representing a chemical bond are used and categorized after their bond type. The main problem of representing a molecular structure by a 3D representation is the accessibility of accurate 3D structures and the fact that for a reasonable representation many energetically comparable conformers would need to be considered in the representation of a molecule.

The most advanced and computing-intensive molecular representations are molecular graphs. In molecular graphs the chemical structure of a molecule is directly utilized and used as an architecture for a neural network. Each atom of the molecule is described by a node of the resulting neural network and each bond is described by a connection between those nodes. Therefore, each layer of the resulting neural network contains as many nodes as atoms are present in the molecule and those layers are interconnected accordingly to the bonds in the molecule. After the assignment of initial states to every node, each atom receives inputs from the previous atom layer and updates its own state. By building the resulting neural network deeper, each node is interconnected with more nodes and is therefore obtaining more information about a larger part of its chemical environment and the molecule. This means that the representation of molecular structures is achieved by training one neural network per compound and the real representation is learned through machine learning. To obtain a chemically interpretable output of the molecular graphs a read out function which is also often realized by another neural network is used.

This work will focus on the usage of 2D representations of molecules in the form of gray scale images and will compare and evaluate their performance in regression and clustering tasks on the MoleculeNet, ESOL Dataset which includes water solubility and SMILE strings of 1128 organic compounds. For the prediction of the water solubility of organic compounds several artificial intelligence tools have been developed. Some of them use regression equations like the general solubility equation (GSE) to predict the water solubility of organic drug-like compounds and achieved impressive accuracies with a root mean square error (RMSE) of 0.62. Nevertheless, the use of the GSE requires the measured octanol water partition coefficient as well as the measured melting point of a compound to provide reliable results. Other tools use previously calculated topological indices like E-states to describe molecular structures in combination with a neural network and achieved a RMSE of 0.67 for fixed sizes of the training and validation set. Even Monte Carlo simulations of solutes in water can be used to obtain regression equations which are able to predict the water solubility of organic compounds with a RMSE of 0.7. In the literature there are also water solubility prediction accuracies of human experts available who achieved an impressively low RMSE of 0.94. Flagship of the solubility prediction algorithms for drug-like compounds are computational-intensive molecular graph based methods achieving RMSE as low as 0.58.

In contrast to the more advanced and not directly interpretable representation methods used by other algorithms, 2D representations of molecules as used in this work are widely used to predict molecule properties in teaching and chemical research. A trained chemist can easily derive molecule properties by analyzing its structure and functionalities. To predict, for example, the water solubility of an organic compound a chemist would intuitively count the number of polar groups, consider the size of the molecule, and estimate the water–molecule interaction by counting the number of H-bond donors and acceptors. All of these variables are influencing the water solubility and empirical rules, such as Lipinski’s “Rule of Five,” are taking these into account.

Therefore, 2D molecule representations and their directly accessible features should provide enough information to predict the molecule’s structure-dependent properties.

To validate this thesis different types of artificial neural networks (ANN) will be evaluated on their performance in a water solubility regression-and a clustering-task. For the clustering a combination of a convolutional autoencoder and UMAP will be used for dimension reduction, followed by labeling via HDBSCAN. The ESOL water solubility regression task will be performed using five different approaches:

1. A dense neural network (DNN)
2. A convolutional neural network (CNN)
3. A CNN and a DNN trained on augmented data
4. A neural network pretrained by transfer learning on GDB17
5. A neural network trained on four structure derived chemical descriptors

Thereby a conceptually simple method which can easily be implemented by a chemist who is not explicitly trained in programming while yielding a prediction accuracy comparable to other more complex approaches is presented. Using unique 2D representations of molecular structures in combination with a successfully applied transfer learning approach is a straightforward process and the generation of 2D molecular representations is a common and computing power inexpensive step. In contrast to the implementation of a two-step pipeline which first computes molecular fingerprints or descriptors for every compound and then feeds them into a ML algorithm. The direct use of molecular structures in form of common 2D-representations is a streamlined and desirable process.
Figure 1. The pre-processing process. The SMILE-strings are converted via the RDKit-environment to quadratic RGB-images of the molecular formula. The 3D RGB-images are then converted to gray scale images and the pixel values scaled to the range of 0 to 1.

The resulting clusters will be critically evaluated by their chemical usefulness. The results of the regression task will be compared to the MoleculeNet[16] benchmark results. Furthermore, the influence of the ANNs size and the image size will be discussed.

2. Experimental Section

2.1. Datasets

All regression tasks were performed with the ESOL dataset published by John S. Delaney[17] and curated by MoleculeNet.[16] The used dataset contains among other information the water solubility of 1128 organic small molecules in the form of the logarithmic maximum dissolved concentration (log(sol)). All compounds are represented as SMILE strings and contains on average 13 heavy atoms, has a molecular weight of 203.5 g mol\(^{-1}\) and a logarithmic solubility of \(-3.11\). Moreover, the log(sol) distribution follows nearly a Gaussian bell distribution.

All transfer learning approaches were realized with the GDB-17 dataset[29] created by Reymond et al. This dataset contains SMILE string representations of 50 million organic small molecules and no molecule descriptors or labels.

2.2. Pre-Processing

All molecules in the MoleculeNet[16] ESOL dataset are represented by SMILE strings. The SMILE strings were converted to quadratic RGB-images via the Python RDKit-environment.[30] To reduce the dimensionality of the images the images were converted to gray scale images via OpenCV.[31] The gray scale images were then converted into NumPy[32] arrays and the pixel values scaled to the range of 0 to 1 by dividing the array by 255. The pre-processing process is illustrated in Figure 1.

2.3. Clustering

The clustering was performed on 28 × 28 pixel images and therefore on 784 dimensional inputs. To reduce the dimensionality of the input images an autoencoder was trained by using a transfer learning approach.[33] Transfer learning approaches are a common way of pretraining and presetting weights of deep neural networks to achieve high accuracies on small datasets. To achieve this, a large available dataset as the GDB-17 dataset[29] with similar features to the original, smaller dataset is used to pretrain the desired network without using the original data. After sufficient training, the pretrained deep neural network can then be further trained on the original dataset leading to higher accuracies and less overfitting.[33] The autoencoder consists of 12 layers, using convolutional layers for feature extraction and pooling layers for dimension reduction. An 84% reduction of dimensions was realized by utilizing three sequential blocks of convolution and pooling layers. Each block consisted of one 3 × 3 convolution layer followed by a 2 × 2 MaxPooling layer. The first block contained a 16 nodes convolution layer. The second and third block contained an 8 nodes convolution layer each. After dimension reduction to 128 dimensions, each picture was resized to 784 dimensions for a comparison to the input image. This was done by using three blocks of 3 × 3 convolution layers and 2 × 2 Up-Sampling layers. The architecture of the autoencoder is shown in Figure 2.

The obtained autoencoder was trained on 54 000 molecules (validation split 0.1) on the GDB-17[29] dataset for 600 epochs with
Figure 2. Architecture of the autoencoder used for the dimension reduction of 28 × 28 pixel images for clustering pre-processing.

Figure 3. DNN and CNN architectures used to evaluate the influence of image size and ANN size on the solubility prediction accuracy.

an Adagrad-optimizer\[34\] resulting in a binary cross entropy loss value of 0.26. Further dimension reduction was implemented using UMAP\[25\] decreasing the number of dimensions from 128 to plottable 2D outputs. The obtained outputs were then clustered via HDBScan\[26\] and the resulting clusters were plotted in a colored scatter plot.

2.4. CNN/DNN Comparison

To evaluate the performance of different sized DNNs and CNNs on various sized quadratic images multiple ANNs were built. The architecture of the ANNs is illustrated in Figure 3. The influence of image size was analyzed for three different DNNs and two different CNNs on three different image sizes (14 × 14, 28 × 28, 56 × 56). The influence of the ANN size was evaluated by varying the number of nodes in the dense $n_d$ or the convolutional layers $n_{conv.1}$, $n_{conv.2}$.

Each DNN consisted of a flatten layer to reshape the quadratic (p, p, 1) shaped images to a 1D tensor. The number of nodes in the hidden Dense layer was varied from $n_d = 32$ to $n_d = 128$ and the layer was L2-kernel regularized\[35\] to prevent overfitting. All models were using the Adam optimizer\[36\] and mean square error (MSE) loss function.\[37\] Activation of the dense layers was realized via the ReLU-activation function.\[38\]

The CNNs consisted of two sequential 3 × 3 convolution layers with varying number of nodes ($n_{conv.1} = 32, 64$), ($n_{conv.2} = 16, 32$). A flatten layer was added to reshape the resulting multidimensional output into a 1D tensor before feeding it to the output layer.

2.5. Data Augmentation CNN/DNN

The influence of data augmentation\[39\] was evaluated for a DNN and a CNN model. The architecture of those models is shown in Figure 4. Augmentation of the dataset was done by randomly flipping the images horizontally, vertically, and rotating them in a range from 0° to 15°. Therefore, each image was only fed one time through the ANN and was randomly changed before each epoch. All dense layers in the ANN are L2-kernel regularized\[35\] and activated via the ReLU-activation function.\[38\] To prevent overfitting, additional dropout layers\[40\] were added with a dropout rate of 0.5. The augmented CNN is equal to the CNN ($n_{conv.1} = 64$ and $n_{conv.2} = 32$) shown before.
Figure 4. CNN and DNN architecture used to evaluate the influence of data augmentation on the solubility prediction accuracy.

Figure 5. Architecture of the pretrained molecular weight predicting model (left) and the subsequently constructed solubility predicting model (right). Pretrained layers are colored blue, truncated layers pale blue, and newly added layers gray. The top eight layers of the pretrained model were truncated and replaced by four untrained layers.

2.6. Transfer Learning Model

To improve the CNN prediction accuracy a deep CNN model was constructed via transfer learning. For pretraining of the model the GDB17 dataset\(^{29}\) was used. The model was pretrained on 360,000 compounds, predicting the exact molecular weight. After being trained for three training episodes ([60,000 images, 40 epochs], [120,000 images, 40 epochs], [180,000 images, 18 epochs]) a mean absolute error of ±3.65 g mol\(^{-1}\) was reached. The last eight layers of the pretrained model were truncated and substituted by a flatten layer, followed by a dense layer, a dropout layer, and a single node dense layer. The resulting model was trained for 100 epochs on 80% of the ESOL dataset\(^{17}\) and validated on 10%. The architecture of the pretrained model and the solubility prediction model are shown in Figure 5.

2.7. Model Trained on Expert Structure Derived Features

The last and smallest model was built by manually preselecting important features accessible by visually analyzing the molecule structure. The input consisted of four features: molecular weight, number of hetero atoms, number of H-bond acceptor groups,
and the number of H-bond donor groups. All the mentioned features should have an impact on the water solubility and are considered in Lipinski’s “Rule of five.”[23] The architecture of the model is shown in Figure 6.

Scaling of the input data was achieved by dividing each feature by its maximum value. Therefore, each feature is in the range of 0 to 1. All dense layers, except the output layer, are L2-kernel regularized[35] and activated via the ReLU-activation function.[38] The resulting model was trained for 4000 epochs on 80% of the ESOL dataset[27] and validated on 10%.

3. Results and Discussion

3.1. ESOL Clustering

The scatter plot of the dimension reduced data is shown in Figure 7. All 1128 compounds of the ESOL dataset are represented as colored dots accordingly to their assigned cluster. In total, 16 various sized clusters were identified, and their chemical usefulness evaluated. It is to mention that the cluster assignment itself is an unsupervised process and that therefore no labels are required.

Clusters far apart from each other are structurally more diverse then clusters in direct neighborhood. Small and condensed clusters like cluster 7 and 16 are more likely to be structurally uniform in contrast to large and widely spread clusters such as cluster 9. Cluster 0 and 1 are separated from the rest of the clusters and in direct vicinity to each other and should therefore be uniform among themselves and be structurally different to the rest of the clusters. The same trend is observable for clusters 2 and 3 in a less distinct manner.

Example compounds of the four distinct clusters (0, 5, 2, 15) are shown in Figure 8. Example compounds of cluster 9, a widely spread cluster, are also provided to examine their difference.

As illustrated in Figure 8, compounds assigned to cluster 2 are mainly linear hydrocarbon chains with few hetero atoms and only small, unbulky side chains. Cluster 2 is in direct neighborhood to cluster 3 and should therefore be structurally similar. Cluster 0 contains mainly 5 and 6 membered carbocycles with a 1- or 1,3,5-substitution pattern. Cluster 5 contains sphere shaped molecules, mainly unsubstituted 5 and 6 membered carbocycles or heterocycles. Also compounds similar in 2D-shape, like condensed ring systems, are observable. Cluster 15 was merged with cluster 16, containing larger cyclic systems, such as condensed carbocycles. Additionally, carbon chain linked bicyclic systems were labeled as cluster 15. The widely spread cluster 9 contains primarily Y-shaped molecules and is therefore only 2D-structural sensible. The containing molecules are very diverse in structure and only uniform in their 2D-shape. This illustrates a major drawback in using images as inputs for chemical clustering, yielding clusters similar in 2D-shape but not in chemical motifs or substructures.

Chemical descriptors and statistics of all assigned clusters are given in Table 1, which contains the mean molecular weight, the mean number of hetero atoms, the mean number of aromatic cycles, and the mean number of cyclic systems of each cluster. For every variable the standard deviation is given to evaluate its uniformity. The chosen descriptors are not part of the clustering process itself and only allow to evaluate the results of the unsupervised clustering process.

As depicted in Table 1, the clustering is well suited for the differentiation of substituted benzenes. Therefore, the clusters 1, 4, 7, and 11 contain benzenes of different substitution patterns and with different sized substituents. The distinguishing of linear and cyclic systems is also possible and even condensed aromatic and condensed saturated cyclic systems can be distinguished (clusters 13, 12, and 15). This leads to a low standard deviation and cluster uniformity as listed in Table 1.

Larger deviations in the number of aromatic systems or the number of cyclic systems can be explained by similar visual shape. As an example: The upper right compound of Label 0 in Figure 8 resembles the shape of the monosubstituted cyclic systems, therefore it is assigned to cluster 0, but contains in fact eight cyclic systems. Those wrong assignments have a major impact on the standard deviation for small clusters. The large deviation in the number of hetero atoms is conspicuous and it seems that the clustering does not involve the number of hetero atoms for the differentiation of the compounds.

Also problematic are the three clusters (8, 9, and 14) which are not chemically sensible. These clusters are only similar in 2D-shape and not in substructures or chemical motifs. Therefore, they show no usefulness in chemical contexts.

3.2. Influence of the Image Size on DNN/CNN Accuracy

To evaluate the influence of the image size on the water solubility prediction accuracy different sized images (14 × 14, 28 × 28, 56 × 56, 128 × 128) for the DNN architecture and (14 × 14, 28 × 28, 56 × 56) for the CNN architecture were used. All DNN
models were trained for 100 epochs, all CNN models for 40 epochs, resulting in almost converged loss and test mean absolute error (MAE) values. Using different sized images resulted in trained models of varying accuracy. It is shown that the assumption that larger images lead to a more accurate DNN prediction is wrong. In fact, an optimal size of $28 \times 28$ pixel images was observed by evaluating the mean absolute and mean squared errors of the test set. The influence of image size on test set MAE, loss function, and number of parameters for the DNN architectures are shown in Figure 9.

Considering the loss function (mean squared error) of the training set, it is shown that for increasing image size and increasing $n_j$, the MSE of log(sol) decreases. The intersection of the curve $n_j = 128$ and $n_j = 64$ could be explained by statistical fluctuations but the general trend is more significant.

A different influence of image size and varying minima are shown for the test set MAE. The global minimum of the test set MAE (1.3) is reached on an image size of $28 \times 28$ pixels and the largest $n_j$ of $n_j = 128$ nodes. For both models, $n_j = 128$ and $n_j = 64$, the test set MAE is increasing with increasing image size after passing their local minima.

The trend of the test set MAE and the loss function could be explained by the quadratic increase of the number of parameters with image size and therefore the rapidly increasing entropic capacity of the DNN models. The extended number of parameters leads to a larger capacity for the model to memorize the training set and impairs the generalization ability.[27] Only if the number of possibly obtainable information of the images would scale in comparable manner to the number of parameters, the extension of image size would positively affect the predication accuracy.

The influence of layer size was further analyzed by increasing the number of nodes $n_j$ of the best performing DNN from 128 to 1024, resulting in a test MAE of only 1.2. Therefore, it seems that the influence of $n_j$ is not as significant as the influence of the image size.

The influence of the image size on the test set MAE and the loss function for the CNN models is shown in Figure 10. For the influence of the image size on the training set loss function a similar trend as for the DNNs is observable. Increasing image size, as well as increasing $n_{\text{conv}_1}$ and $n_{\text{conv}_2}$ leads to a decrease of the loss value. In comparison to the DNNs, the test set MAE decreases with increasing CNN capacity and image size as well.

Figure 7. Clusters obtained by dimension reduction via autoencoder, UMAP ($n_{\text{neighbors}} = 20$, min_dist = 0.05) and labeling via HDBScan ($\text{min}_\text{cluster\_size} = 18$, $\text{min}_\text{samples} = 12$). Each cluster is represented by its own color. In total, 16 different sized clusters are labeled. Black dots are compounds not assignable to any cluster (noise).
Figure 8. Example compounds of the distinct clusters 2, 0, 5, and 15. Compounds of cluster 9 are shown to illustrate the difference between dense, narrow sized clusters, and widely spread clusters. Label 2 = “linear hydrocarbon chains,” Label 0 = “5- and 6-membered carbocycles with 1- and 1,3,5-substitution pattern,” Label 5 = “monocyclic, unsubstituted cyclic systems and condensed carbocycles,” Label 15 = “large condensed carbocycles and hydrocarbon chain linked bicyclic systems,” Label 9 = “Y-shaped molecules.”

Table 1. All assigned clusters and their visual description as well as chemically relevant descriptors.

| Cluster (n_{comp}) | Visual description                  | M_{mol} [g mol\(^{-1}\)] | \(\eta_{\text{Ar sys}}\) | \(\eta_{\text{cyc sys}}\) | \(\eta_{\text{stat atom}}\) |
|-------------------|------------------------------------|-----------------------------|--------------------------|--------------------------|-----------------------------|
| 0 (48)            | Carbo cyclic with 1- or 1,3,5-substitution pattern | 161.12 ± 137.7               | 0.75 ± 0.3               | 1.35 ± 1.4               | 2.67 ± 3.0                  |
| 1 (47)            | Benzenes 1,4- or 1,2,4,6-substitution pattern | 166.94 ± 60.2               | 0.83 ± 0.4               | 1.02 ± 0.1               | 2.77 ± 1.9                  |
| 2 (79)            | Hydrocarbon chains, partially branched | 141.60 ± 64.6               | 0.05 ± 0.2               | 0.08 ± 0.4               | 2.41 ± 2.3                  |
| 3 (113)           | Linear hydrocarbon chains, few het. atoms | 128.47 ± 67.9               | 0.02 ± 0.2               | 0.04 ± 0.2               | 1.37 ± 1.6                  |
| 4 (36)            | Benzenes with 1,2-substitution pattern | 142.85 ± 43.2               | 1.22 ± 0.6               | 1.47 ± 0.5               | 1.63 ± 1.3                  |
| 5 (21)            | Monocyclic unsubstituted systems (circular shape) | 116.88 ± 69.4               | 1.33 ± 2.0               | 1.86 ± 1.8               | 0.42 ± 0.7                  |
| 6 (55)            | Unsaturated hydrocarbon chains, het. atom rich | 180.04 ± 126.6              | 0.44 ± 0.8               | 0.67 ± 1.3               | 3.38 ± 3.3                  |
| 7 (23)            | Benzenes with 1,3- or 1,2,4,5-substitution pattern | 163.94 ± 67.8               | 0.96 ± 0.2               | 1.00 ± 0.0               | 2.47 ± 1.6                  |
| 8 (34)            | Cross like shaped molecules (chemically not sensible) | 190.63 ± 97.5               | 0.79 ± 1.0               | 1.21 ± 1.4               | 3.01 ± 2.5                  |
| 9 (134)           | “Y” shaped molecules (chemically not sensible) | 216.30 ± 92.6               | 1.13 ± 1.0               | 1.52 ± 1.3               | 4.17 ± 2.7                  |
| 10 (24)           | Bulky group with hydrocarbon chain substituent | 270.90 ± 101.6              | 1.30 ± 0.7               | 1.63 ± 1.0               | 5.00 ± 2.9                  |
| 11 (22)           | Monosubstituted benzenes with large substituents | 135.17 ± 45.7               | 0.95 ± 0.4               | 1.09 ± 0.3               | 1.32 ± 1.2                  |
| 12 (24)           | Bicyclic systems with hydrocarbon linkers | 264.38 ± 95.2               | 1.79 ± 1.0               | 2.16 ± 0.9               | 3.79 ± 2.6                  |
| 13 (82)           | Condensed aromatic ring systems | 229.00 ± 63.8               | 2.06 ± 1.1               | 2.55 ± 0.9               | 3.22 ± 2.8                  |
| 14 (59)           | Triangle shaped molecules (chemically not sensible) | 240.16 ± 104.2              | 1.02 ± 1.0               | 1.27 ± 1.1               | 5.17 ± 3.2                  |
| 15 (69)           | Large, saturated, condensed carbocycles | 334.02 ± 65.17              | 0.69 ± 0.9               | 3.14 ± 1.1               | 5.20 ± 2.5                  |
Nevertheless, the best performing CNN (test_MAE = 1.46) is outperformed by the best performing DNN (test_MAE = 1.30). Evaluating the influence of the image size on the computing time, the trivial correlation between an increasing number of parameters and the computing time can be validated. The comparison between an image size of 14 × 14, 28 × 28, 56 × 56, and 128 × 128 pixels for the DNN shows an increase of computing time of up to 7.5 times of the smallest computing time value. The DNN performs the regression task for the smallest image size (14 × 14) with a computing time of 2 ms per step compared to 15 ms per step for the largest image size (128 × 128). Nevertheless, the highest accuracy of the DNN on 28 × 28 pixel images is reached with a computing time of 3 ms per step. The computing time for the CNN exceeds the computing time of the DNN on all image sizes. The fastest computing time of 6 ms per step was reached for an image size of 14 × 14 pixels. An 11.8 times higher computing time of 71 ms per step was reached for the biggest evaluated image size of 56 × 56 pixels. This demonstrates the underlying more demanding computational tasks of the CNN in comparison to the DNN.

Considering the huge success of image classification CNNs as ImageNet[41] the surpass of the CNN is conspicuous and could be caused by the small number (902) of available training compounds of the ESOL data set. To enhance the performance of the
CNN and to eliminate the lack of available training compounds, alternative approaches as data augmentation and transfer learning were studied.

3.3. Data Augmentation as Compensation for Small Datasets

Data augmentation for the DNN model resulted in a slight improvement of the prediction accuracy (test_MAE = 1.28 compared to 1.30). The augmented DNN model was also able to learn for more epochs (100 vs 2000). Considering the minor improvement of the log(sol) prediction accuracy and longer computing time, data augmentation seems infeasible for this model and task. Evaluation of the CNN model and its performance with augmented data showed no improvement in solubility prediction accuracy. The solubility prediction accuracy even decreased by implementing data augmentation (test_MAE = 1.61 compared to test_MAE = 1.46). Figure 11 illustrates the training curve and validation curve of the augmented DNN and CNN.

The improvement of the DNN could be explained by its stronger feature orientation dependency and the increasing number of learnable features by rotating and flipping of the training images. The minor improvement of the prediction accuracy may suggest that the orientation of chemical structures only plays a smaller role in feature extraction from images.

Considering the more orientation independent feature extraction ability of CNNs, it is understandable that the prediction accuracy did not improve. However, the increase of the test set mae for the CNN could be explained by untypical representations of the training images. The compound images were all produced by the same RDKit algorithm and the augmented training data set may be too different to the unaugmented test set, leading to an increased test MAE.

Therefore, data augmentation in this scenario leads to no major improvements of the solubility prediction accuracy and a transfer learning approach was implemented.

3.4. Transfer Learned CNN

Creating a deep CNN without preselected weights and trained layers can be a challenge for small datasets. Transfer learning is one approach to avoid collecting more data and achieving high performance models with few data. A deep, exact molecular mass predicting CNN was trained using the GDB-17\(^{[23]}\) dataset. The top layers were replaced by untrained dense layers and the obtained model was trained on 80% of a random split ESOL dataset. The lowest test_MAE of 0.87 for logarithmic water solubility predictions was achieved after 100 epochs. The log(sol) prediction error distribution and the predicted/true value-water solubility scatter plot is shown in Figure 12.

The most common prediction error is around 0 and the errors are ranging from \(\approx-4\) to 3. The shape of the error distribution resembles a Gaussian bell distribution. Therefore, most of the predicted values almost corresponded to the true log(sol) values and outliers negatively affecting the test set MAE.

3.5. Expert Selected Feature Model

The last and simplest model was trained on chemist preselected features important for water solubility. The input tensor consisted of the molecular mass, the number of hetero atoms, and the number of H-bond donor and acceptor functionalities. Consequently, only four scaled values (between 1 and 0) were used to represent each compound. The built DNN only consisted of 33 total trainable parameters and was trained, in under 15 min, on a random split ESOL dataset (80/10/10, train/val./test) for 4000 epochs on a laptop CPU, in contrast to the Tesla K80 GPU used before. The lowest test set MAE achieved was 0.78, demonstrating the importance of expert supervision. The simplest DNN, trained on preselected and for a chemist visually obtainable chemical features, outperforms the complex transfer learned CNN. The results of the expert selected feature model are shown in Figure 13.

Fewer outliers are observable for the scatter plot of the expert selected feature model in comparison to the transfer learning model. However, the most common prediction error shifted from around 0 to \(\approx-0.5\) and a second common prediction error of 1 is noticeable.

In this context the significance of expert supervision for machine learning is shown, demonstrating the role of ML.
algorithms as an expert tool, gaining their full potential when combined with expertise.

4. Conclusion

It was shown that 2D-images of molecules, as used by chemists for decades, are feasible for machine learning tasks. The combination of a transfer learned CNN and 2D-molecule representations resulted in a model that can compete with other conventional methods but is less accurate in the prediction of water solubility of organic compounds than more advanced graph-based methods.\[16\]

Furthermore, the relevance of expert supervision was demonstrated by creating a very simple DNN (33 total parameters) trained on expert preselected, visually accessible features. This model was able to outperform the more complex CNN in computing time and prediction accuracy.

An overview over all discussed results is given in Table 2 supplemented by results of the MoleculeNet ESOL benchmark.

![Figure 12](image1.png)

**Figure 12.** Scatter plot of the transfer learned model predicted log(sol) versus true log(sol) (left) and the distribution of the solubility prediction errors (right).

![Figure 13](image2.png)

**Figure 13.** Scatter plot of the expert selected feature model log(sol) predictions versus true log(sol) values (left) and the distribution of the log(sol)-prediction errors (right).

| Model/Lit.                          | MAE   | RMSE |
|------------------------------------|-------|------|
| Best performing DNN                | 1.30  | 1.64 |
| Best performing CNN                 | 1.46  | 1.81 |
| Augmented data DNN                 | 1.28  | 1.28 |
| Augmented data CNN                  | 1.61  | 2.00 |
| Transfer learned CNN               | 0.87  | 1.17 |
| Expert selected feature DNN         | 0.78  | 1.04 |
| MoleculeNet benchmark conventional methods | /     | 0.99\[16\] |
| MoleculeNet benchmark graph-based methods | /     | 0.58\[16\] |

Table 2. Summary of all discussed water solubility prediction accuracies, supplemented by MoleculeNet benchmark results.
Acknowledgements

This work was performed as part of the Cluster of Excellence Fuel Science Center (EXC 2186) funded by the Excellence Initiative by the German federal and state governments to promote science and research at German universities.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

ESOL, machine learning, solubility, transfer learning

Received: July 6, 2020
Revised: August 10, 2020
Published online: September 6, 2020

[1] D. L. Mobley, K. L. Wymer, N. M. Lim, J. P. Guthrie, J. Comput. Aided. Mol. Des. 2014, 28, 135.
[2] G. Montavon, M. Rupp, V. Gobre, A. Vazquez-Mayagoitia, K. Hansen, A. Tkatchenko, K. - R. Müller, O. A. von Lilienfeld, New J. Phys. 2013, 15, 095003.
[3] A. Dubey, Int. J. Mol. Biol. 2018, 3, 23.
[4] J. Devillers, D. Domine, C. Guillon, W. Karcher, J. Pharm. Sci. 2015, 104, 2862.
[5] D. C. Elton, Z. Boukouvalas, M. D. Fuge, P. W. Chung, J. Mol. Des. 2019, 4, 828.
[6] J. Lim, S. Ryu, J. W. Kim, J. Cheminformatics. 2018, 10, 31.
[7] M. Stahl, H. Mauser, M. Tsui, N. R. Taylor, J. Med. Chem. 2005, 48, 4358.
[8] D. Weininger, J. Chem. Inf. Comput. Sci. 1988, 28, 31.
[9] S. R. Heller, A. McNaught, I. Pletnev, S. Stein, D. Tchekhovskoi, J. Cheminform. 2015, 7, 23.
[10] B. O. Mitchell, Wiley Interdiscip. Rev. Comput. Mol. Sci. 2014, 4, 468.
[11] A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, G. Pujadas, Methods 2015, 71, 58.
[12] D. Rogers, M. Hahn, J. Chem. Inf. Model. 2010, 50, 742.
[13] M. Rupp, A. Tkatchenko, K. R. Müller, O. A. Von Lilienfeld, Phys. Rev. Lett. 2012, 108, 058301.
[14] D. Duvenaud, D. Maclaurin, J. Aguilera-Iparraguirre, R. Gómez-Bombarelli, T. Hirzel, A. Aspuru-Guzik, R. P. Adams, presented at Proc. of Advances in Neural Information Processing Systems, Montreal, Canada, December 2015.
[15] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, G. E. Dahl, presented at 34th Int. Conf. on Machine Learning, Sydney, Australia, August 2017.
[16] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswingd, V. Pande, Chem. Sci. 2018, 9, 513.
[17] J. S. Delaney, J. Chem. Inf. Comput. Sci. 2004, 44, 1000.
[18] Y. Ran, S. H. Yalkowsky, J. Chem. Inf. Comput. Sci. 2001, 41, 354.
[19] V. Tetko, V. Y. Tanchuk, T. N. Kasheva, A. E. P. Villa, J. Chem. Inf. Comput. Sci. 2001, 41, 1488.
[20] W. L. Jorgensen, E. M. Dufﬁ, Bioorg. Med. Chem. Lett. 2000, 10, 1155.
[21] S. Boobier, A. Osbourn, J. B. O. Mitchell, J. Cheminformatics. 2017, 9, 63.
[22] A. Lusci, G. Pollastri, P. Baldi, J. Chem. Inf. Model. 2013, 53, 1563.
[23] C. A. Lipinski, J. Pharmacol. Toxicol. Methods 2000, 44, 235.
[24] M. A. Kramer, AIChE J. 1991, 37, 233.
[25] L. McNees, J. Healy, J. Melville, arXiv:1802.03426 2018.
[26] R. J. G. B. Campello, D. Moulavi, J. Sander, in Advances in Knowledge Discovery and Data Mining. PAKDD (Eds: J. Pei, V. S. Tseng, L. Cao, H. Motoda, G. Xu), Springer, Berlin, Heidelberg 2013
[27] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning MIT Press 2016.
[28] C. Shorten, T. M. Khoshgoftaar, J. Big Data 2019, 6, 60.
[29] L. Riguttigke, R. Van Deursen, L. C. Blum, J. L. Reymond, J. Chem. Inf. Model. 2012, 52, 2864.
[30] G. Landrum, RDKit: Open-source cheminformatics.
[31] G. Bradski, Dr. Dobb’s Journal of Software Tools, 2000.
[32] T. E. Oliphant, A guide to NumPy, USA: Trelgol Publishing, 2006.
[33] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, in Artificial Neural Networks and Machine Learning (Eds: V. Korkova, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis), Lecture Notes in Computer Science, ICANN 2018.
[34] J. Duchí, E. Hazan, Y. Singer, presented at 23rd Conf. on Learning Theory, Haifa, Israel, June 2010.
[35] C. Cortes, M. Mohri, A. Rostamizadeh, presented at 25th Conf. on Uncertainty in Artificial Intelligence, Montreal, Canada, June 2009.
[36] D. P. Kingma, J. L. Ba, presented at 3rd Int. Conf. on Learning Representations, San Diego, CA, May 2015.
[37] W. James, C. Stein, in Breakthroughs in Statistics (Eds: S. Kotz, N. L. Johnson), Springer, New York 1992, p. 443.
[38] P. Ramachandran, B. Zoph, Q. V. Le, presented at 8th Int. Conf. on Learning Representations, Vancouver, Canada, April 2018.
[39] I. Cortes-Ciriano, A. Bender, J. Chem. Inf. Model. 2015, 55, 2682.
[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, J. Mach. Learn. Res. 2014, 15, 1929.
[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, Int. J. Comput. Vis. 2015, 115, 211.