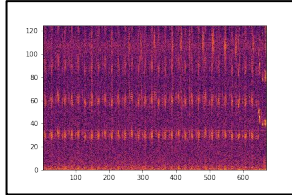
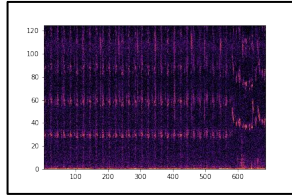


Bat Orientation Calls

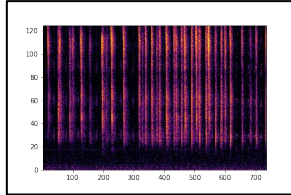
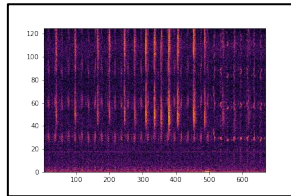
Gaillinger, Ott, Bittl

- 4118 Spektrogramm-Bilder von Fledermäusen
- 9 verschiedene Fledermäuse

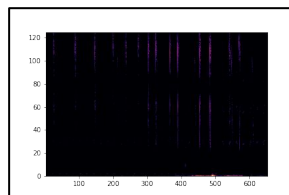
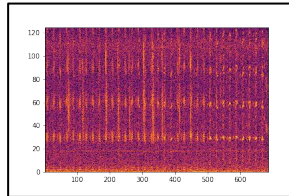
Bechsteinfledermaus



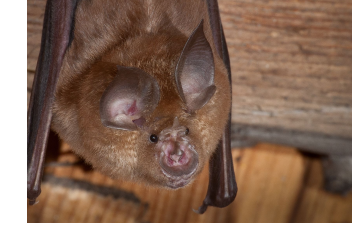
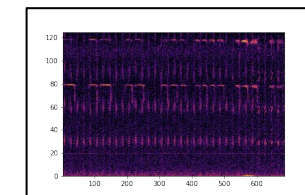
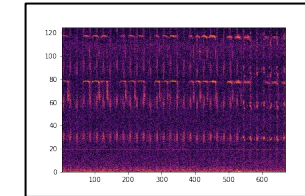
Bartfledermaus



Mausohr

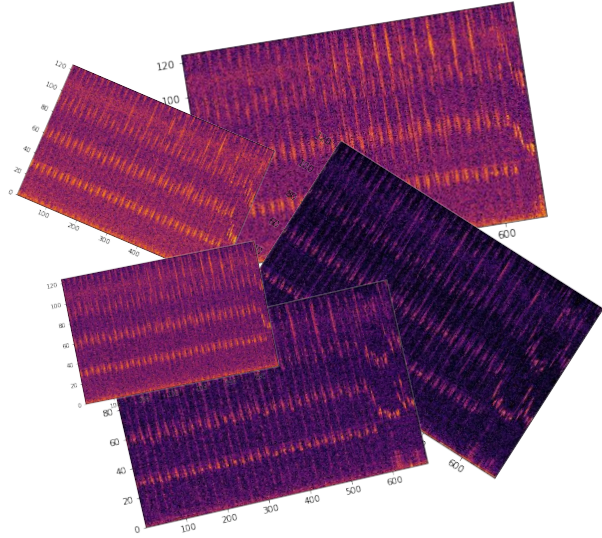


Große Hufeisennase

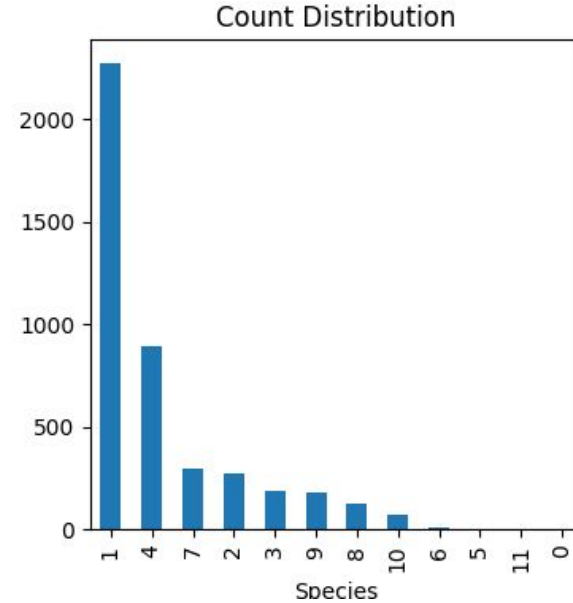


Datenimport & Aufbereitung

Von den Bildern zu etwas Nutzbarem

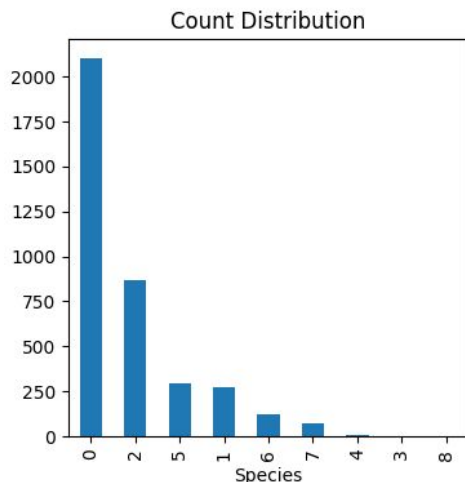


4118 Originale Bilder
(432x288)



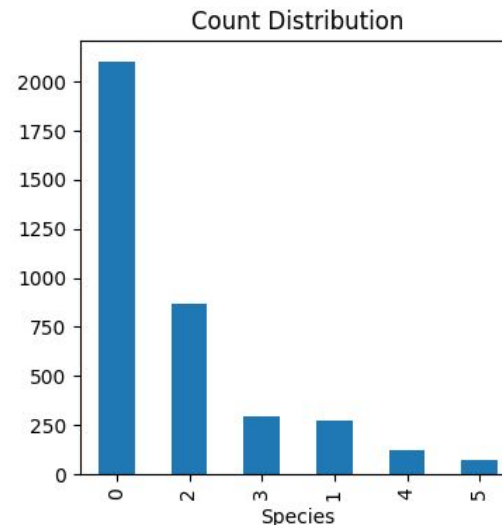
12 verschiedene Labels,
4310 insgesamt

- Daten enthalten u.a. 40 Duplikate, ein falschen Namen (“&Mausohr” statt “Mausohr”), 180 Schwarzbilder und 180 unbestimmbare Fledermäuse



- Nur noch 9 Klassen

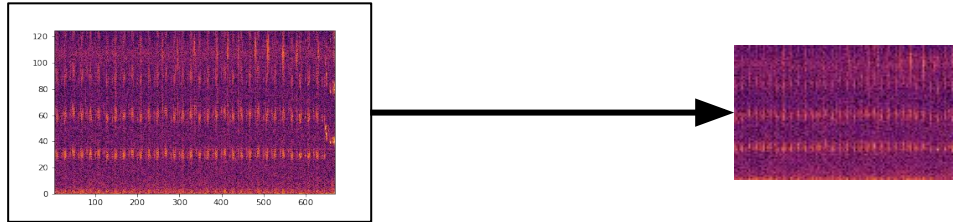
- trotzdem unbrauchbare Klassen: teilweise nur 1-7 Samples pro Klasse



- Verbleiben 3734 Samples

Datenvorverarbeitung:

- Kategorische Labels in Numerische Labels wandeln
- Farbwerte zwischen 0-1 normalisieren (reduzierte Konvergenzzeit)
- Ist er denn aber wirklich nutzbar?
 - **Nein!** Bilder mit 432x288 zu groß! (Modell rechnet mit 128GB RAM auf AWS...)
 - Lösung: PCA? In Chunks splitten? Bilder auf 128x82 skalieren!
- Abschneiden des transparenten Randes und der Skala => abschneiden

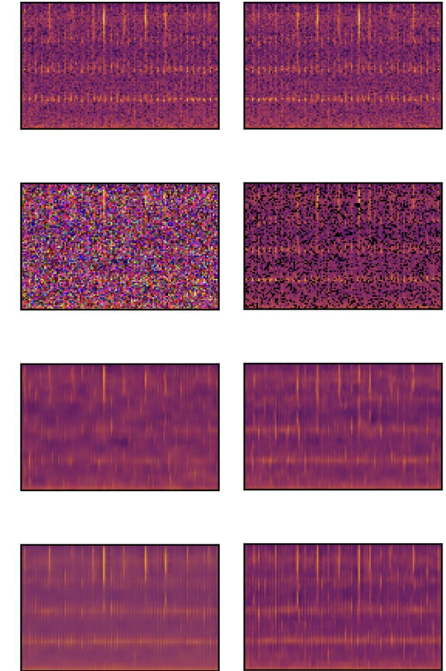
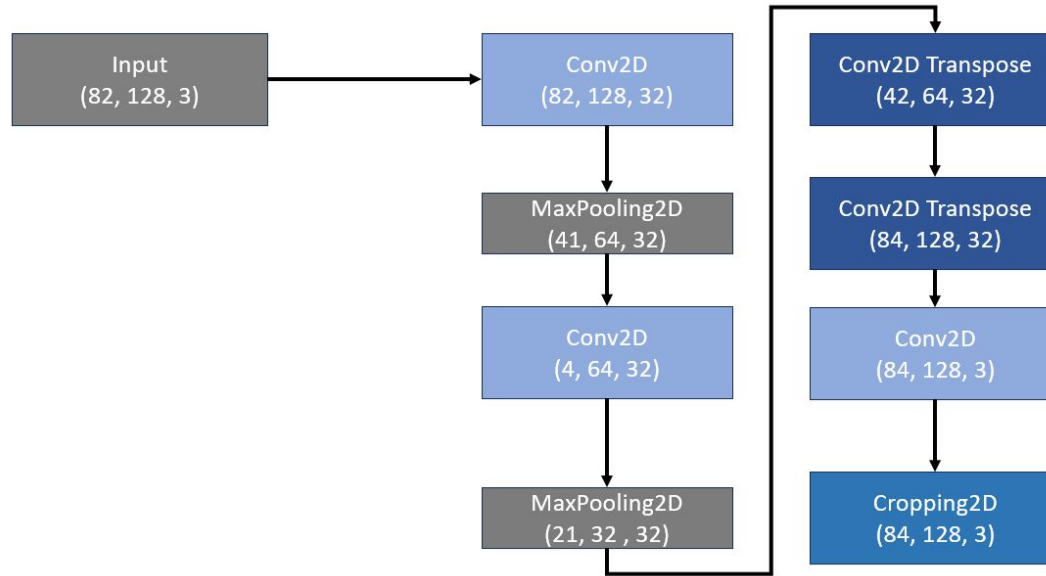


- Sehr stark ungleiche Klassenverteilung resampling der Daten mit ADASYN/SMOTTEEN

Convolutional Autoencoder zum Denoisen

=> Vorher Salt&Pepper oder Gaussian Noise angewendet damit der Autoencoder das Denoisen erlernen kann

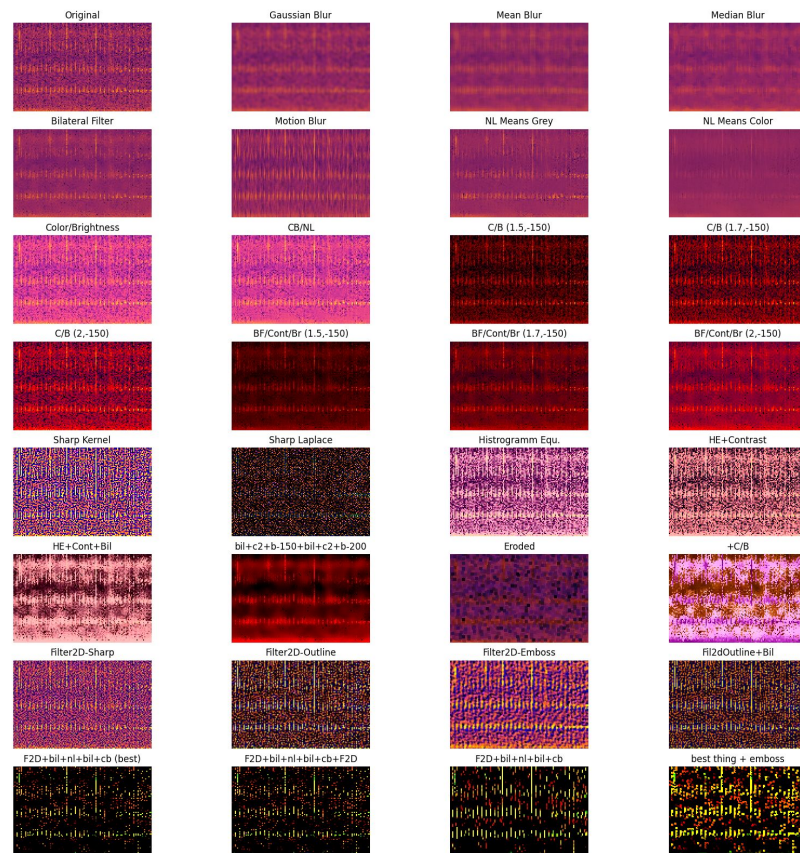
=> Letztendlich nicht besser als ohne Autoencoder (ca. 78%)



Datenoptimierung:

- Durch Filter Inputbilder verbessern
- Kontrast & Helligkeit, Bilateral Filter und fastNLMMeansDenoising mit am besten
- Bestes Ergebnis ca. 91% Test Acc. auf CNN:
 - Outline Detection Conv2D Filter + Bilateral Filter + fastNLMMeansDenoising + Bilateral Filter + Contrast(*1.7) + Brightness(-150)

Frage: Werte mit geresampten Testdaten
=> Ist das ein Problem?



Problematik bei resampling der Testdaten

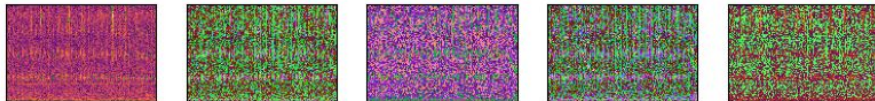
Um den Ernstfall zu simulieren sollten keine synthetischen Bilder in den Testdaten sein.

ABER:

Entfernen der geresamplen Testdaten führt zu einem drastischen Rückgang der Accuracy

-> resample Daten weichen vermutlich zu stark von den Originaldaten ab

-> Modell hat sehr stark die künstlich erstellten Bilder gelernt



Hinweis: Uns ist dieser Fehler erst recht spät aufgefallen, weshalb wir nicht mehr alle Filter ausprobieren konnten. Die besten haben jedoch keine Vorteil gebracht.

=> Wahrscheinlich hat Resampler die Filter gelernt

Ergebnisse auf dem CNN:

Durchführung	Avg. Test Acc
Kein Resample, top 6 classes (> 60), cnn	79%
Kein Resample, top 4 classes (>200), cnn	83%
Kein Resample, top 2 classes (>500), cnn	87%

Train & Validation Resample (Smotteen(all,enn=Mode,n=3)), top 6 classes (> 60), cnn	63%
Train & Validation Resample (Smotteen(all,enn=Mode,n=3)), top 4 classes (>200), cnn	80%
Train & Validation Resample (Smotteen(all,enn=Mode,n=3)), top 2 classes (>500), cnn	87%

Train & Validation Resample (Smotteen(all,enn=Mode,n=7)), top 6 classes (> 60), cnn	68%
Train & Validation Resample (Smotteen(all,enn=Mode,n=7)), top 4 classes (>200), cnn	79%
Train & Validation Resample (Smotteen(all,enn=Mode,n=7)), top 2 classes (>500), cnn	86%

Train Resample (Smotteen(all,enn=Mode,n=3)), top 6 classes (> 60), cnn	73%
Train Resample (Smotteen(all,enn=Mode,n=3)), top 4 classes (>200), cnn	81%
Train Resample (Smotteen(all,enn=Mode,n=3)), top 2 classes (>500), cnn	88%

Classification Trees

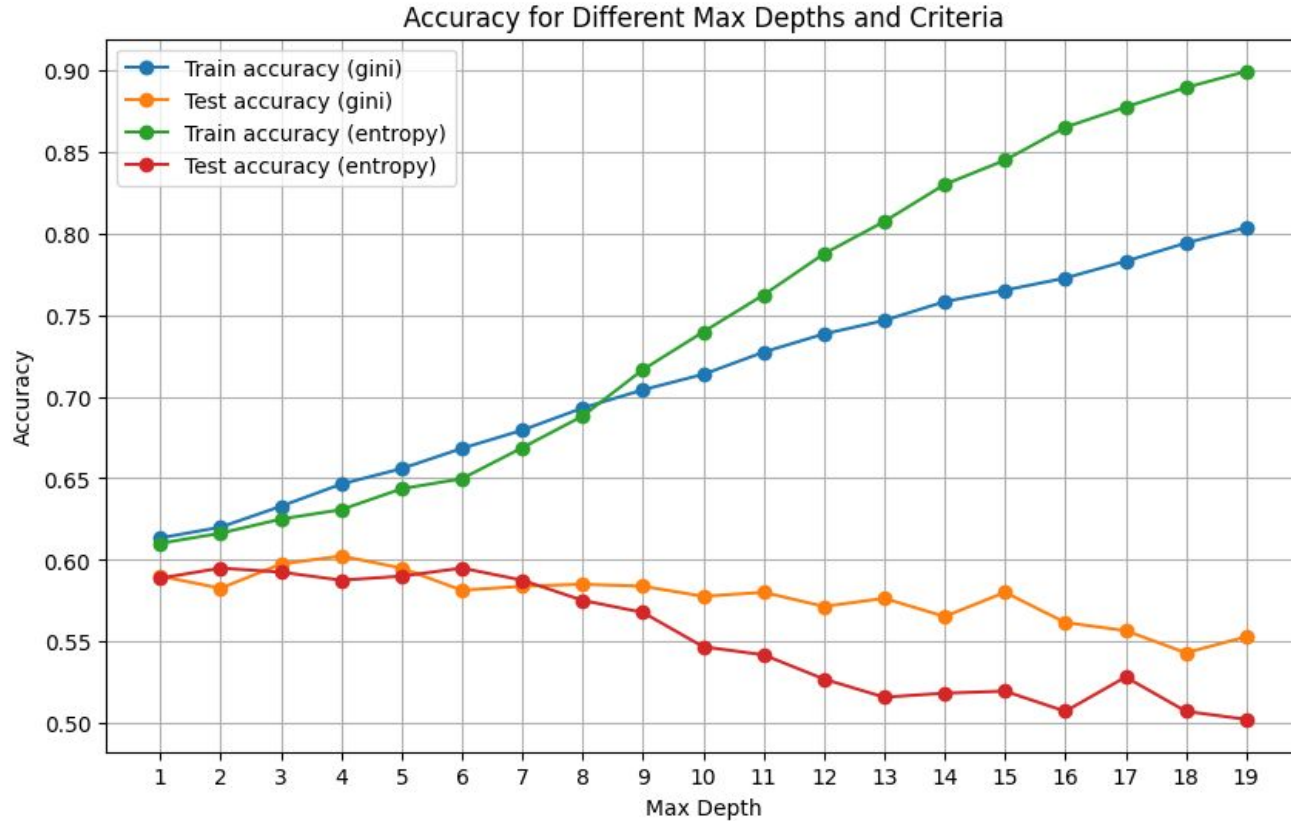
Klassifizieren mit Bäumen und Wäldern

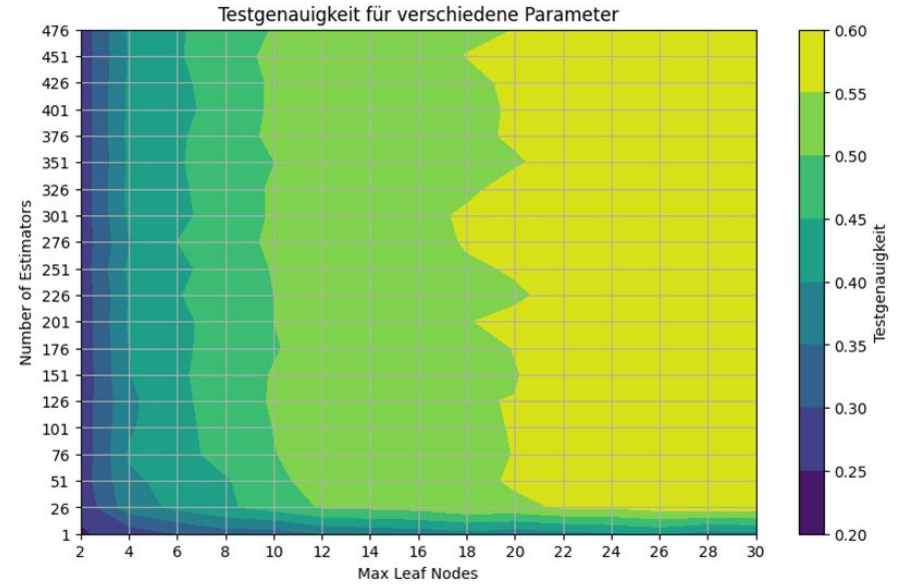
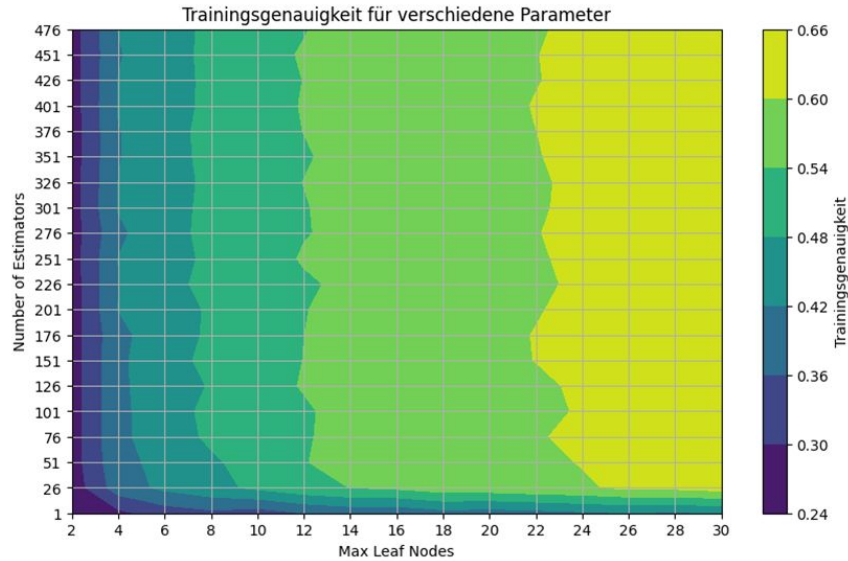
Grundsätzliche Problematiken bei Classification Trees

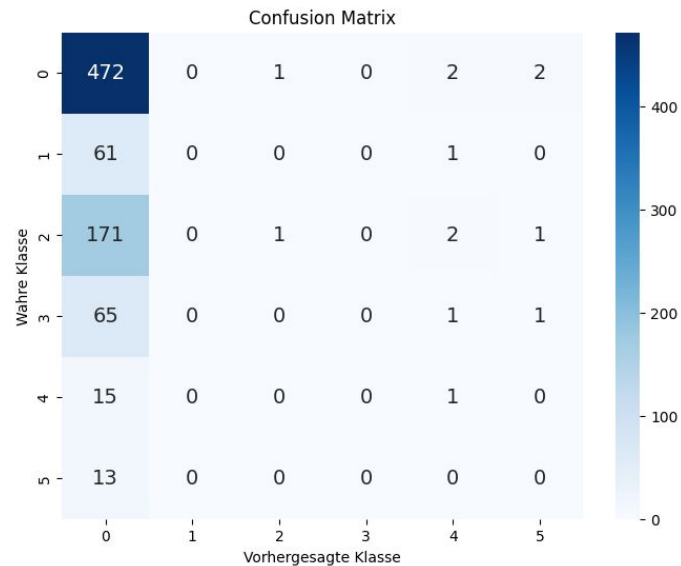
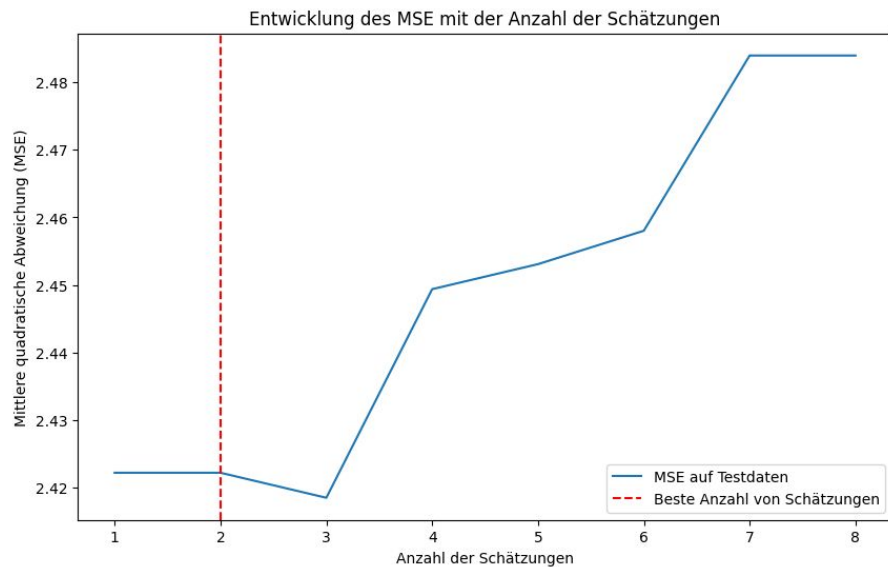
Flattening der Daten

- flache 2D-Struktur
- Bezug zwischen benachbarten Pixeln geht verloren
=> Struktur der Spektrogramme geht verloren
- schwierig richtigen Split zu finden, da Pixel alleine wenig aussagekräftig

=> sehr schlechte Qualität der Entscheidungsbäume





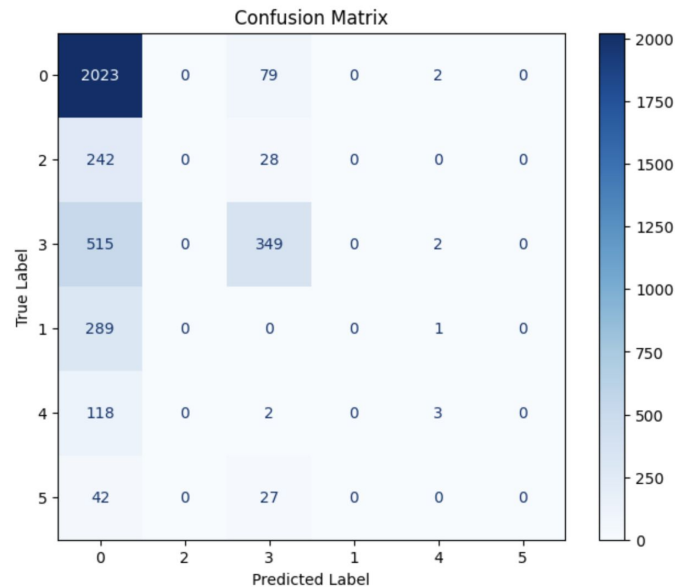
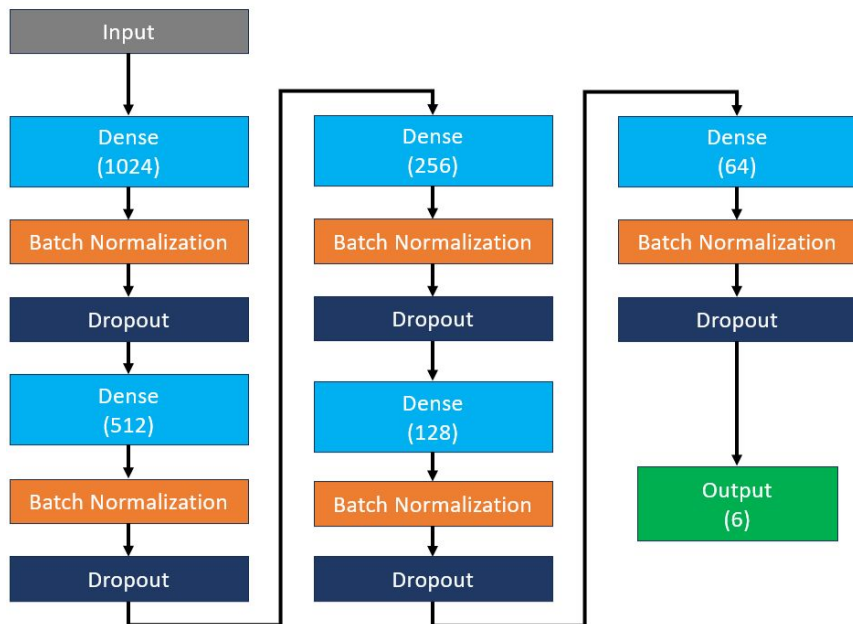


FFNN & CNN

Neuronale Netze zur Rettung

Grundsätzliche Problematik bei FFNN:

Wie bei den Trees werden die Daten als Vektor rein gegeben, weshalb die Zusammenhänge zwischen nebeneinander liegenden Pixeln verloren gehen.



Test Accuracy: ~63%

Convolutional Neural Network

- Stratified 10-Fold-Cross Validation (Class-Distribution wird in den Splits beibehalten => bei ungleich verteilten Daten gut)
- Weight Decay (Durch lineare ReLu's unproblematisch)
- Pruning
- Batch Normalization (normalisiert den Batch nach einem Layer)

Optimizer Wahl

Durchführung	Avg. Test Acc
SGD	77%
SGD_exp	72%
ADAM	79%
ADAM_exp	74%

exp = ExponentialDecay der Learning Rate

Model:

epochs = 200 (EarlyStopping)

batch_size = 32,

dropout_rate = 0.2,

weight_decay_alpha = 0.01

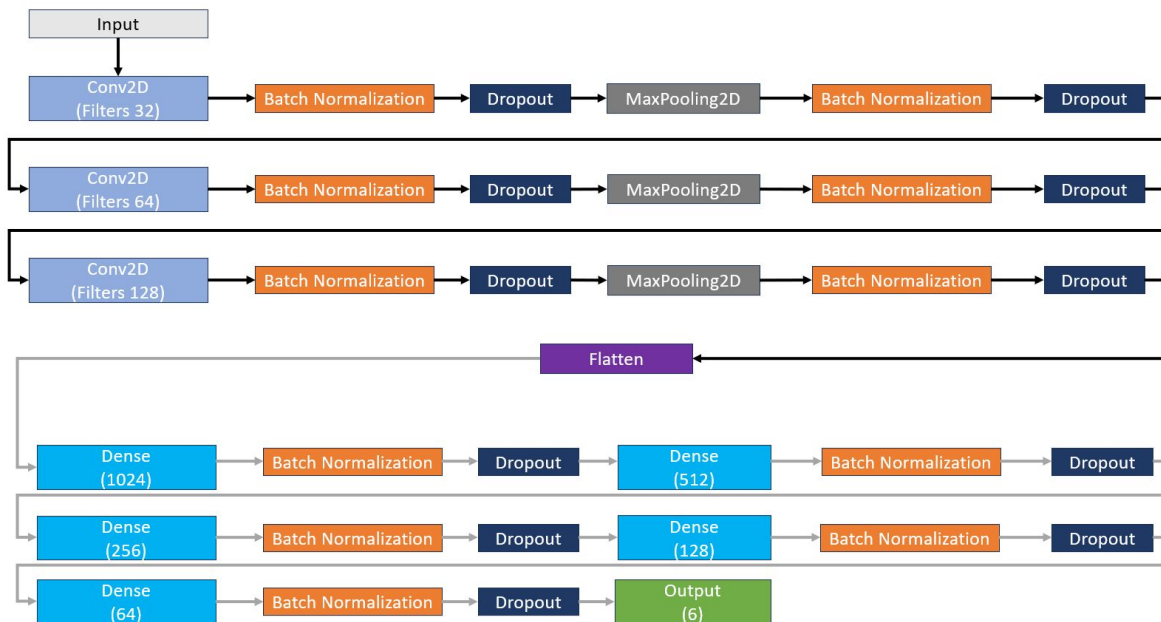
kernel_size = (3, 3)

pool_size = (2, 2)

padding = "same"

activation_func = ReLu (außer
Output 'softmax')

loss_func = Sparse Categorical
Crossentropy



Ergebnisse:

VGG19 und ResNet50 als groben Vergleich
(nur Klassifikationslayer nachtrainiert)

VGG19

Input Layer: 1
Convolutional Layers: 16
Klassifikationslayer: 2
Output Layer: 1

ResNet50

Input Layer: 1
Convolutional Layers: 48
Klassifikationslayer: 2
Output Layer: 1

Durchführung

Avg. Test Acc

VGG19

Kein Resample, top 6 classes (> 60), vgg19	82%
Kein Resample, top 4 classes (>200), vgg19	83%
Kein Resample, top 2 classes (>500), vgg19	91%

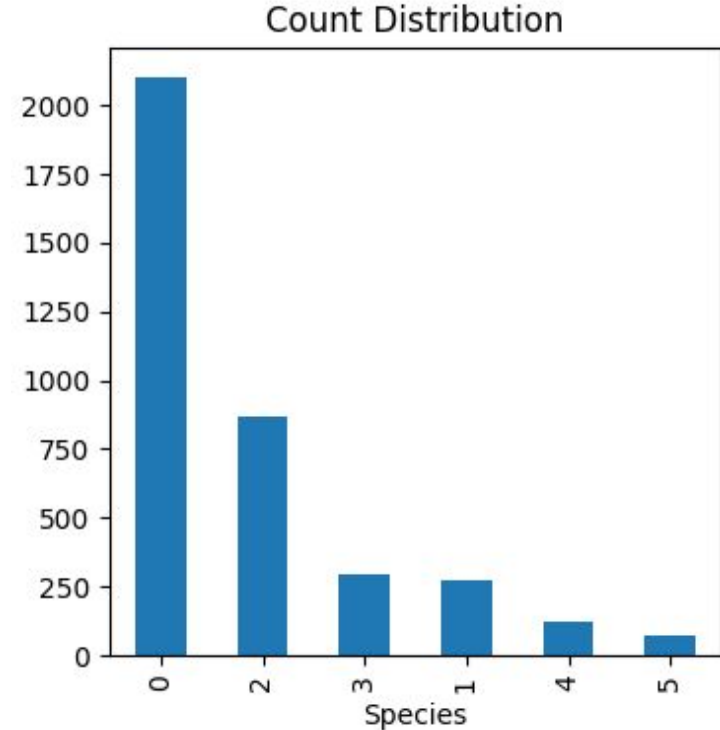
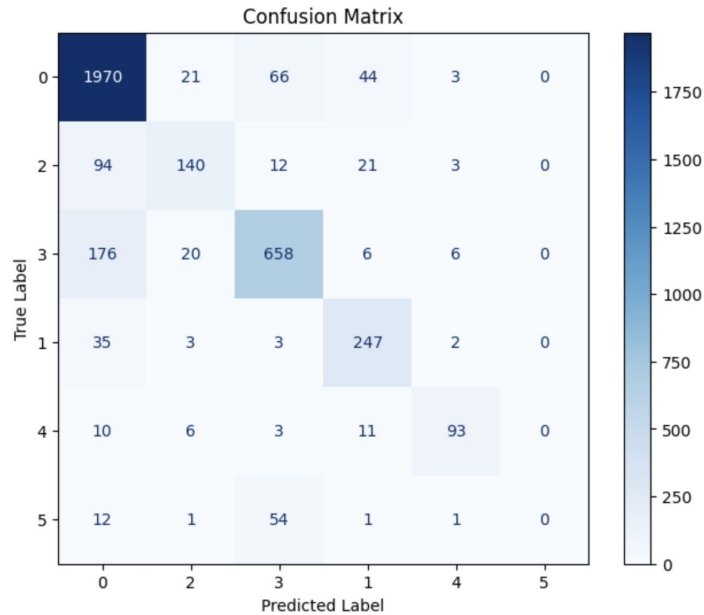
ResNet50

Kein Resample, top 6 classes (> 60), resNet50	70%
Kein Resample, top 4 classes (>200), resNet50	74%
Kein Resample, top 2 classes (>500), resNet50	84%

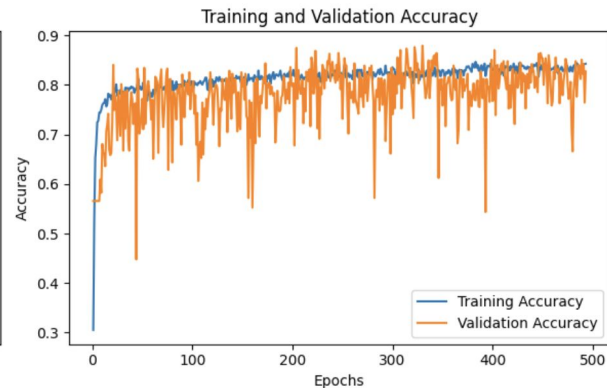
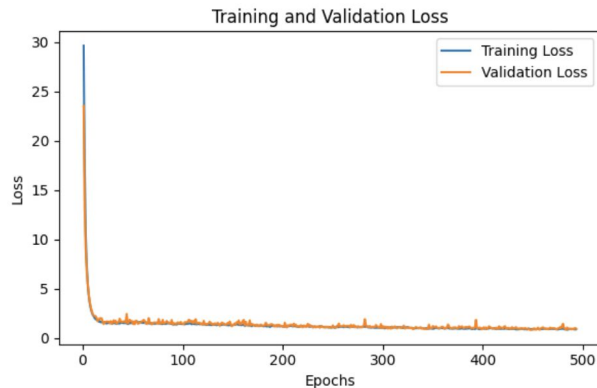
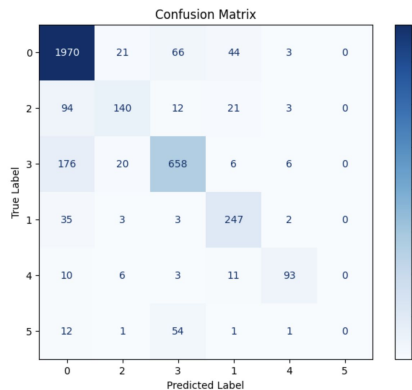
Unser Modell

Kein Resample, top 6 classes (> 60), cnn	83%
Kein Resample, top 4 classes (>200), cnn	86%
Kein Resample, top 2 classes (>500), cnn	89%

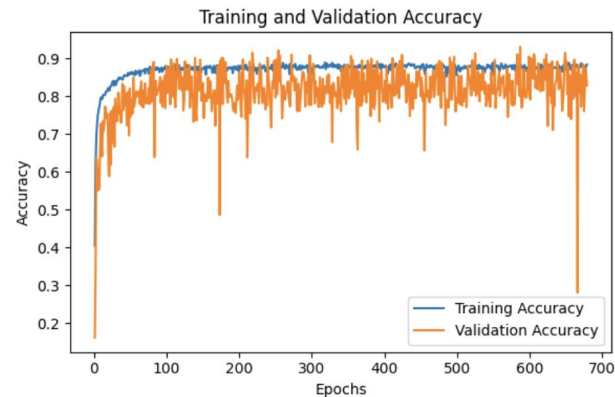
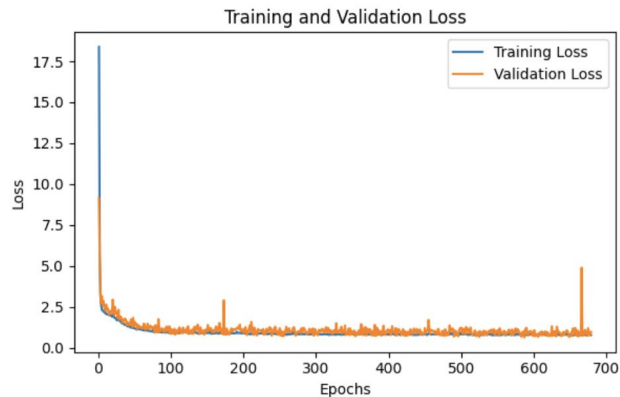
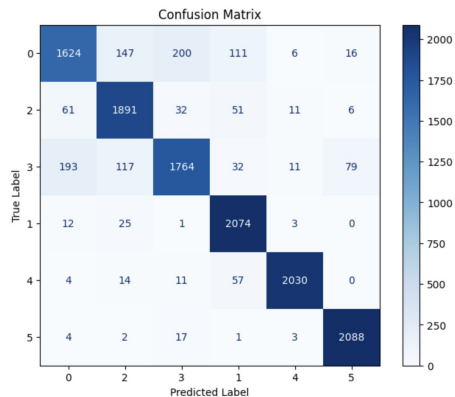
Es bestätigt sich die Annahme, dass die Klassen mit wenig Samples sehr schlecht predicted werden.



ohne Resampling



mit Resampling



Zukunftsausblick

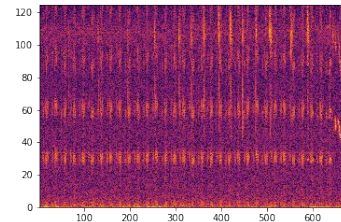
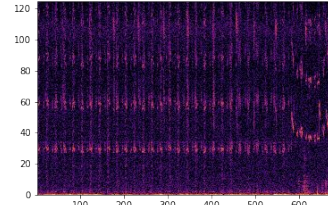
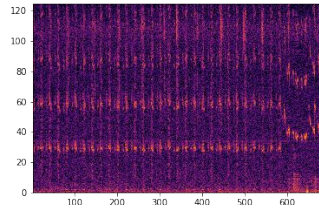
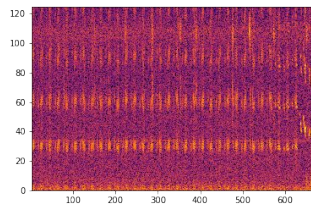
Fazit und weitere Ansätze

Fazit:

- Rechenpower/RAM war ein großes Problem
- Klassenverteilung war der Knackpunkt
- Resampling funktioniert nicht immer so einfach
- Wie erwartet ist CNN für Bilder das beste Model

Ansätze zur Verbesserung:

- Noise effektiv aus den Bildern entfernen
- Mehr echte Daten vor allem für die unterrepräsentierten Daten sammeln
- Synthetische Daten erzeugen, welche sehr nah an den Originaldaten sind (Expertenwissen über Fledermäuse/Spektrogramme wären sehr hilfreich)
- Filter, welche wichtige Teile effektiv hervorheben (Expertenwissen nötig)



Bat Orientation Calls

Noch Fragen?

