March 2021

GEOLOCATION DRIVEN REINFORCMENT LEARNING-POWERED IN TRANSIT HEADSET NOISE CANCELLATION MECHANISM

Anupam Mukherjee

Vibhor Jain

GEOLOCATION DRIVEN REINFORCMENT LEARNING-POWERED IN TRANSIT HEADSET NOISE CANCELLATION MECHANISM

AUTHORS:
Anupam Mukherjee
Vibhor Jain

ABSTRACT

When one is in transit, different kinds of non-stationary noises may arise. Moreover, in different countries non-stationary noise intensity levels may vary significantly from location to location. For example, in a market in India the intensity of noise will be much higher than that found in a market in the United States. Currently, however, artificial intelligence (AI) -driven headsets are trained on a generic noise-related dataset that is used to filter out noise. To address these types of challenges, techniques are presented herein that support an AI-based, state of the art, intelligent, and interactive algorithm that will detect current headset location based on a geolocation tag and invoke the appropriate specialized purposely-built pre-trained flows to cancel or suppress noise.

DETAILED DESCRIPTION

Background noise is annoying, and it occurs everywhere. Such noise becomes most awkward when one is in transit and either speaking over a phone or listening to music. During an important call while in transit, for example, the listeners on the other end of the call may encounter difficulties in listening to the speaker if the speaker travels by public transport or passes through crowded places such as, for example, a market, a bus terminus, a railway station, an airport, etc.

Currently most headphones either come with Active Noise Control (ANC) to reduce the ambient noise or have a multimicrophone driven noise suppression mechanism. Costlier headsets are normally equipped with soundproofing or a digital signal processor (DSP) driven noise separator or suppressor (e.g., single microphone).

With the advent of AI, deep neural networks (DNN) have been designed to more effectively suppress noise (using software). Such an approach employs training to identify stationary and non-stationary noises that are to be filtered.

When one is in transit, different kinds of non-stationary noises may arise. Moreover, in different countries the intensity level of non-stationary noise may vary significantly from location to location. For example, in a market in India the intensity of noise will be much higher than that found in a market in the United States. Similarly, a public transport in a third world country will produce more noise than that in any first world country where a transport may have, for example, soundproofing means.

Presently, all of the AI driven headsets are trained on a generic noise related dataset to filter out noise (outbound and inbound) from a human voice. However, in a multi-party call involving participants from multiple countries, the effectiveness of such headsets is reduced if some of the participants are in transit and passing through noisy locations. Those headsets do not work satisfactorily in all the countries while travelling by public transport or passing through crowded and noisy places. Most of the headset users in the densely populated countries (such as, for example, India, Pakistan, Egypt, Sri Lanka, Nepal, etc.) face this problem while taking a call or listening to music in transit.

As noted previously, different kinds of non-stationary noises may come along with primary speech or sound while in transit. In different countries, the intensity level of the non-stationary noise may vary significantly from location to location even for the same location type. Hence, a generic noise suppression or cancellation algorithm in the headset will not be completely effective in filtering out noise from the speech or sound when the users are employing it in crowded public places like, for example, a market, a railway station, a bus terminus, an airport, etc. or within public transport like, for example, a bus, a train, a metro-rail, a tram, etc. A more advanced, interactive, and intelligent noise suppression or cancellation algorithm is needed in order to accurately filter out the noise in those situations.

To address the types of challenges that were described above, techniques are presented herein that employ two core principles:

1. Identifying the location (e.g., geolocation) of a headset and classifying the type of location (e.g., place, transport, etc.).
2. Applying an intelligent algorithm to decide the most appropriate degree of noise suppression or cancellation (such as, for example, masking) as per the location and location type.

It is quite obvious that the instant problem falls into an interactive class and, therefore, reinforcement learning (RL) is the best fit, which will use the location of the headset as the environment parameter to decide noise filtration strategy accordingly.

Under aspects of the techniques presented herein the algorithm, which was referred to above, may be characterized as comprising, possibly among other things, the following elements:

- Data preparation steps.
- Training phases.
- Runtime flow.

Each of these elements will be described in the narrative and the illustrations that follow.

A first element, comprising various data preparation steps, may include, possibly among other things, the following steps:

- Pre-classifying the types of location based on different noise characteristics and intensity of noise levels (e.g., high, medium, low for environments like a market, a stadium, a railway station, a bus terminus, an airport, a bus, a train, a shuttle, a place of worship, etc.).
- Pre-classifying the countries (and, optionally, the most important cities in a country) as per environmental noise levels (e.g., collected over a predetermined period of time).
- For every group of countries, based on a geolocation tag pre-locate all of the location types (and surrounding areas up to a certain limit) depending upon different levels of environmental noise, the process includes creating a normal distribution of the noise dataset along with mean, median, or mode (whatever makes sense) and a standard deviation (e.g., on the basis of day by day, or day of the week, etc.) for every location type in every group of countries. These values will aid in creating the synthetic noisy sound from the country-location type-noise matrix for training and validation purposes.
- Creating RL training datasets containing the noise suppression or cancellation strategies for the entries in the above datasets.

The strategies can be purposely-built pre-trained flows (or already available entities such as, for example, Krisp or BabbleLabs application programming interfaces (APIs)) which must have the knowledge of the above variations of location based noises and capability to properly filter them (for, as an example, a location-type and environmental-noise-group-of-country based AI noise masking model). Various steps associated with this flow are depicted in Figure 1, below.

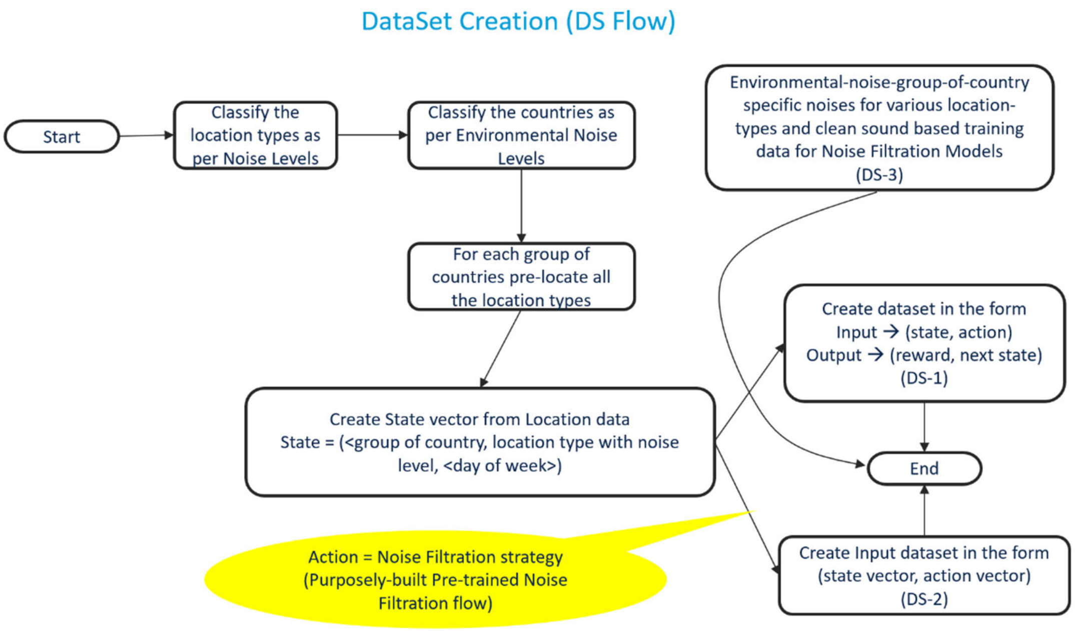
DataSet Creation (DS Flow)
Start
Classify the location types as per Noise Levels
Classify the countries as per Environmental Noise Levels
For each group of countries pre-locate all the location types
Create State vector from Location data
State = (<group of country, location type with noise level, <day of week>)
Action = Noise Filtration strategy (Purposely-built Pre-trained Noise Filtration flow)
Environmental-noise-group-of-country specific noises for various location-types and clean sound based training data for Noise Filtration Models (DS-3)
Create dataset in the form
Input → (state, action)
Output → (reward, next state)
(DS-1)
End
Create Input dataset in the form (state vector, action vector) (DS-2)

Figure 1: Illustrative Data Preparation Steps

For the second element involving training phases, there are two primary models in the proposed RL-based workflow along with enhanced noise masking agents (AI model / available capable 3rd party APIs). One model will be of a multi-label supervised type, which will work as the environment to guide the actual RL model in recommending the most appropriate noise filtration strategy. Another model (internally two DNN models, also known as Double Deep Q-Learning model or Double DQN) will be of RL type, which will interact with the Environment model to decide the best noise suppression/cancellation strategy that will yield the maximum reward. Figures 2A and 2B, below, illustrate example details associated with a DQN architecture.

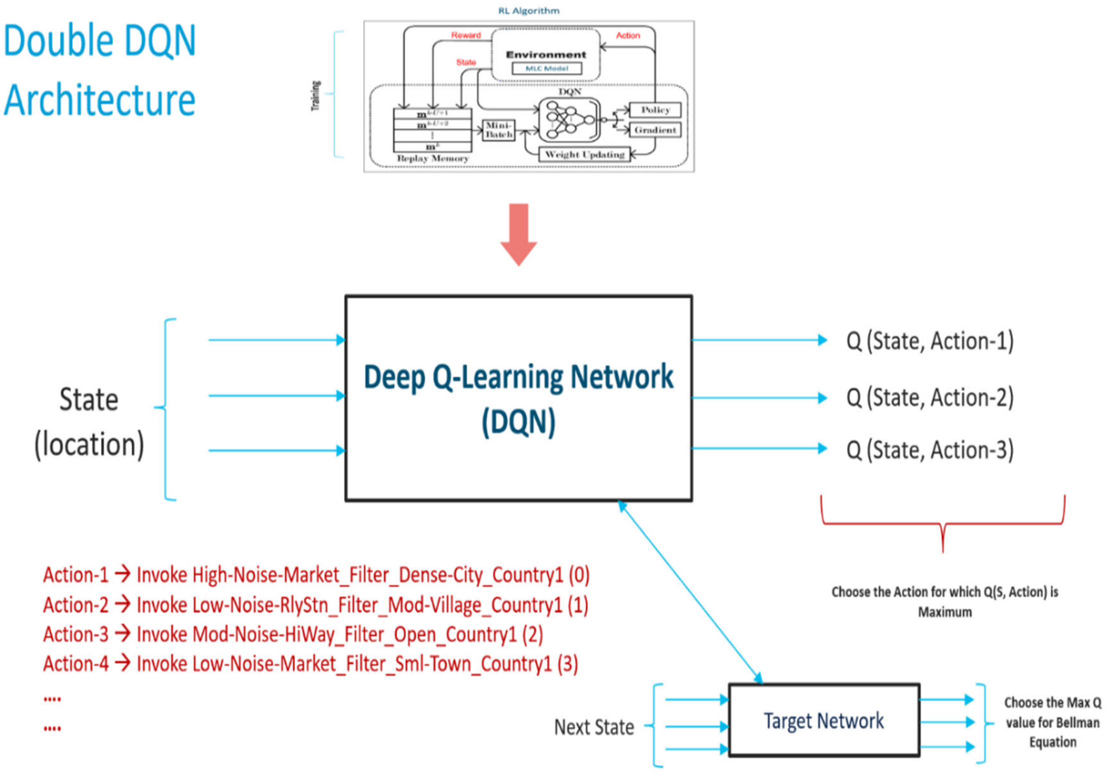
Double DQN Architecture
RL Algorithm
Training
Reward
Action
State
Environment
MLC Model
DQN
Policy
Gradient
Mini-Batch
Weight Updating
Replay Memory
State (location)
Deep Q-Learning Network (DQN)
Q (State, Action-1)
Q (State, Action-2)
Q (State, Action-3)
Choose the Action for which Q(S, Action) is Maximum
Action-1 → Invoke High-Noise-Market_Filter_Dense-City_Country1 (0)
Action-2 → Invoke Low-Noise-RlyStn_Filter_Mod-Village_Country1 (1)
Action-3 → Invoke Mod-Noise-HiWay_Filter_Open_Country1 (2)
Action-4 → Invoke Low-Noise-Market_Filter_Sml-Town_Country1 (3)
....
....
Next State
Target Network
Choose the Max Q value for Bellman Equation

Figure 2A: Illustrative DQN Architecture

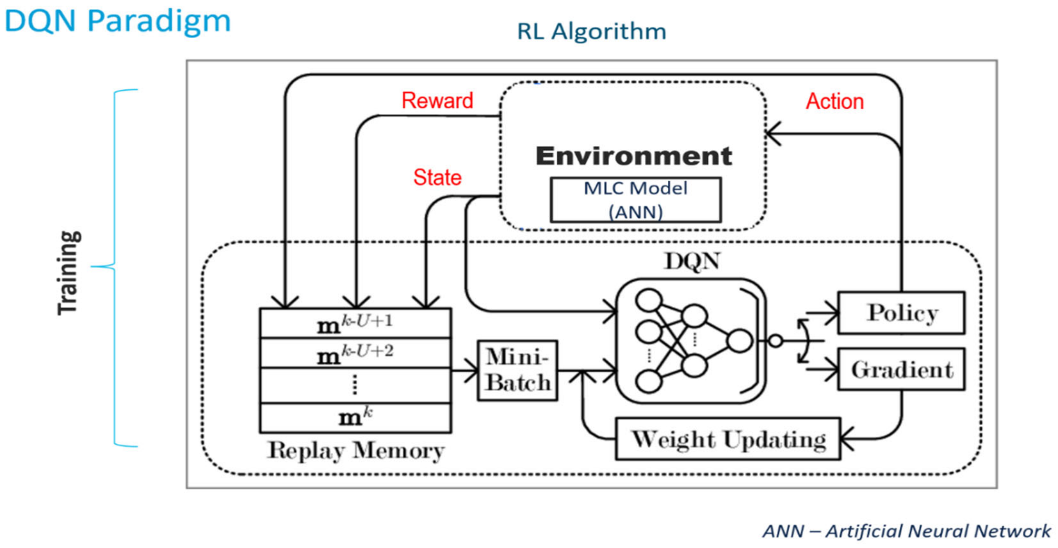

Figure 2B: Illustrative DQN Paradigm of Figure 2A

The models will be trained on their respective datasets. Optionally, there may as well be additional purposely-built AI models (existing models need to be enhanced) which will be trained on the previously described environmental-noise-group-of-country specific various location-type based noises and corresponding clean sound or speech. It should be noted that the difference is that earlier though the training data used to contain various kinds of non-stationary noises, country and respective location based noise characteristics were not emphasized. Hence, the models could not effectively learn the variations of noises from country to country even for the same location types.

The purposely-built pre-trained flows (including any optional AI models or already available entities such as, for example, Krisp or BabbleLabs APIs) must be enhanced to learn the previously described variations of noise based on location and environmental-noise-group-of-country. There must be dedicated flows or APIs corresponding to each environmental-noise-group-of-country. The RL-based noise filtration strategy recommendation flow will then select the most appropriate pre-trained flow with the help of the environmental model in order to filter out the noise from the speech or sound. The proposed sample training flows are illustrated in Figure 3 and 4, below.

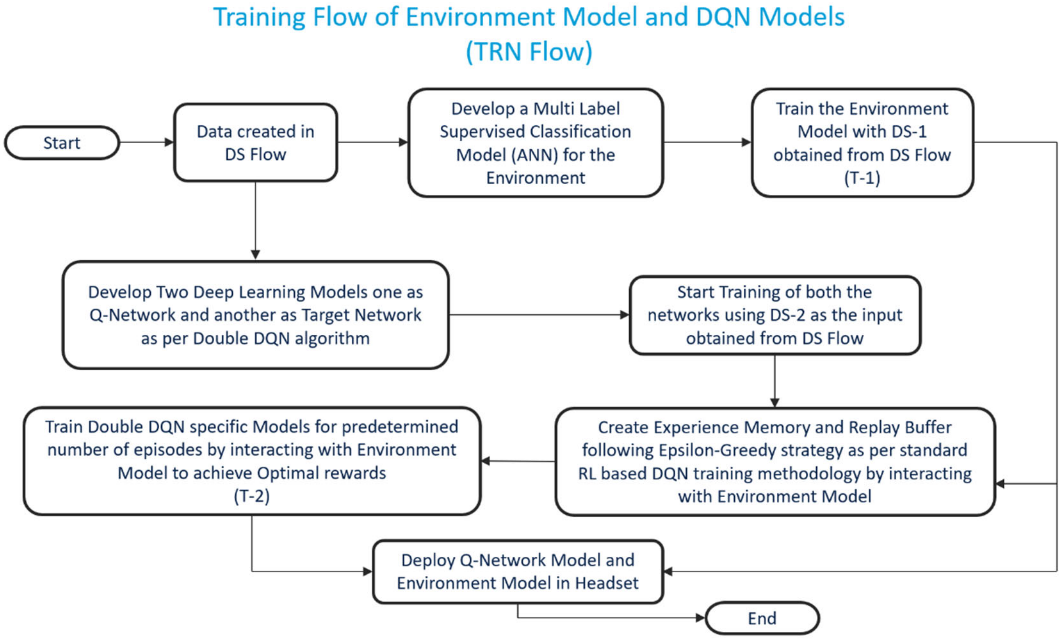
Training Flow of Environment Model and DQN Models
(TRN Flow)
Start
Data created in DS Flow
Develop a Multi Label Supervised Classification Model (ANN) for the Environment
Train the Environment Model with DS-1 obtained from DS Flow (T-1)
Develop Two Deep Learning Models one as Q-Network and another as Target Network as per Double DQN algorithm
Start Training of both the networks using DS-2 as the input obtained from DS Flow
Train Double DQN specific Models for predetermined number of episodes by interacting with Environment Model to achieve Optimal rewards (T-2)
Create Experience Memory and Replay Buffer following Epsilon-Greedy strategy as per standard RL based DQN training methodology by interacting with Environment Model
Deploy Q-Network Model and Environment Model in Headset
End

Figure 3: Illustrative Training Flow Steps

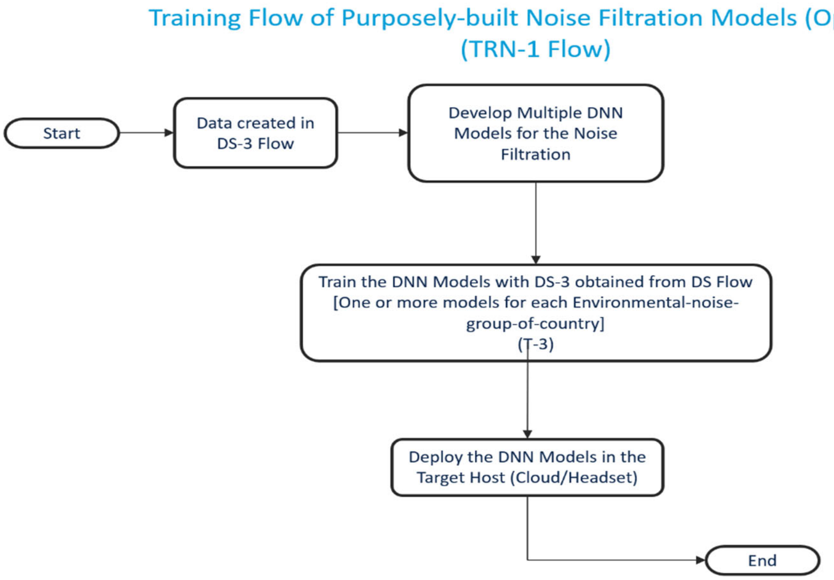
Training Flow of Purposely-built Noise Filtration Models (Optional)
(TRN-1 Flow)
Start
Data created in DS-3 Flow
Develop Multiple DNN Models for the Noise Filtration
Train the DNN Models with DS-3 obtained from DS Flow [One or more models for each Environmental-noise-group-of-country] (T-3)
Deploy the DNN Models in the Target Host (Cloud/Headset)
End

Figure 4: Illustrative Training Flow Steps

For the third element encompassing a runtime flow, headsets will be equipped with a geolocation tracker and the proposed RL-based noise filtration strategy recommender. The proposed sample runtime flow has been illustrated in Figure 5, below. For the purposely-built pre-trained noise filtration flows, there may be two hosting options:

- Option-1: For a cloud based calling solution, the purposely-built pre-trained noise filtration flows will be hosted in the media server on the cloud for maximum scalability.
- Option-2: For all kinds of calling solutions (e.g., cloud, hybrid, on-premise, etc.) the headsets need to be equipped with the appropriate environmental-noise-group-of-country specific purposely-built pre-trained noise filtration flows. The headsets may be provisioned accordingly from, for example, a vendor’s cloud environment based on the location of the user.

The runtime flow may include identifying the current location (via geolocation tagging) of the headset, which will be used to classify the type of location (e.g., place or transport) along with the environmental noise group of the country. Additional information (such as, for example, the day of the week) may optionally be provided for more sophisticated and intelligent noise filtration.

Based on the collected and compiled location data, the pretrained RL model, with the help of pretrained environmental model, will recommend the best noise suppression or cancellation strategy that will yield the maximum reward. The recommended strategy specific purposely-built pre-trained noise filtration flow may be activated (either in the headset or on the cloud) to filter out the noise from both inbound or outbound sound/speech.

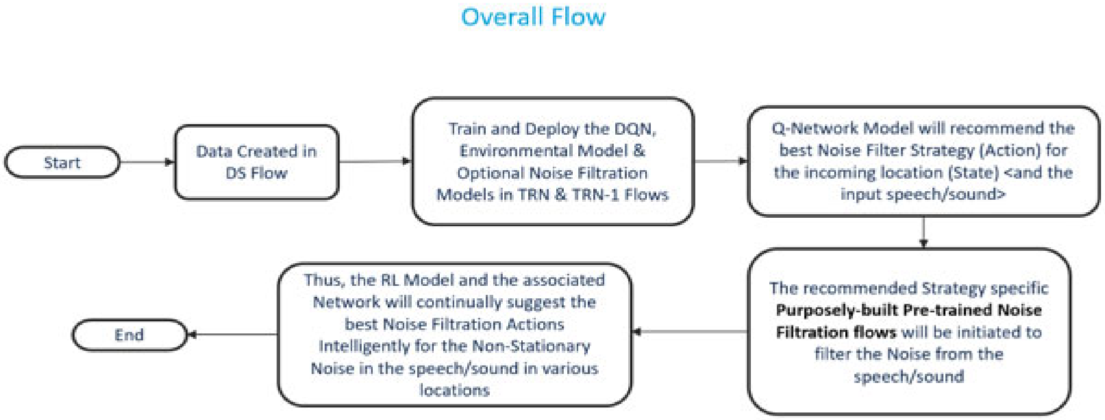
Overall Flow
Start
Data Created in DS Flow
Train and Deploy the DQN, Environmental Model & Optional Noise Filtration Models in TRN & TRN-1 Flows
Q-Network Model will recommend the best Noise Filter Strategy (Action) for the incoming location (State) <and the input speech/sound>
The recommended Strategy specific Purposely-built Pre-trained Noise Filtration flows will be initiated to filter the Noise from the speech/sound
Thus, the RL Model and the associated Network will continually suggest the best Noise Filtration Actions Intelligently for the Non-Stationary Noise in the speech/sound in various locations
End

Figure 5: Illustrative Overall Flow Steps

An illustration of elements of the overall solution architecture, employing aspects of the techniques presented herein, is depicted in Figure 6, below.

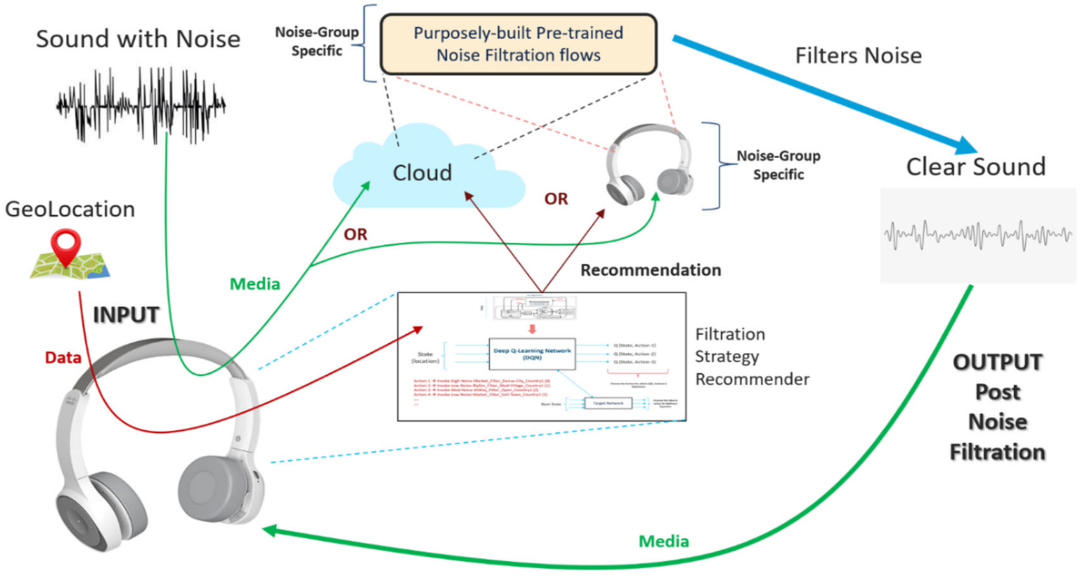
Sound with Noise
Noise-Group Specific
Purposely-built Pre-trained Noise Filtration flows
Filters Noise
Cloud
Noise-Group Specific
Clear Sound
OR
GeoLocation
OR
Media
Recommendation
INPUT
Data
Deep Q-Learning Network (DQN)
Filtration Strategy Recommender
OUTPUT Post Noise Filtration
Media

Figure 6: Overall Solution Architecture

Of interest and note in connection with the techniques that are presented herein, are that, earlier though the training data used to contain various kinds of non-stationary noises, country and respective location based noise characteristics were not emphasized. Thus, the models could not effectively learn the variations of noises from country to country even for the same location types. Under aspects of the techniques presented herein, the previously described purposely-built pre-trained flows (including any optional AI models or already available entities such as, for example, Krisp or BabbleLabs APIs) must be enhanced to learn those variations of noise based on location and environmental-noise-group-of-country. There must be dedicated flows or APIs corresponding to each environmental-noise-group-of-country.

In general, the RL approach's problem is that it has an associated exploration cost, and the initial learning of the environment could degrade the overall user experience. Additionally, different country based location specific noise characteristics complicate the scenarios further. However, under aspects of the techniques presented herein the algorithm that was described and illustrated above along with the purposely-built pre-trained flows help to suppress the initial exploration cost.

Further, for real time noise filtration, the latency must be minimal. Hence, cloud based noise filtration is an obvious choice for the maximum scalability and performance in concurrent voice stream processing. Alternatively, as discussed above in the runtime flow element, environmental-noise-group-of-country specific purposely-built pre-trained noise filtration models can be pushed to the headset from the cloud for minimum latency. However, this may warrant very lightweight and highly optimized AI-based noise filters due to hardware limitations in the headset.

The usage of Krisp or BabbleLabs APIs to filter noise is also an option. However, cloud-based media handling is the most suitable hosting option for the same. Further, the APIs may need to be enhanced to learn the previously described variations of noise based on location and environmental-noise-group-of-country.

If existing AI-based noise filtration mechanisms, instead of the techniques that have been presented herein, are employed to solve the problems that were discussed above, the models need to be trained on the different country based location specific noise characteristics for the accurate suppression or cancellation of the noise from the sound or speech (inbound and outbound). It will complicate the model architecture and increase their size, which in turn will demand more processing power and memory from the hosting platform. Obviously, those models cannot be hosted in the headset due to its limited capability. A cloud approach is the only solution in that case which may increase the latency. The overall dataset will also be quite complicated. Aspects of the techniques presented herein have demonstrated the flexibility in this regard where the purposely-built noise filter models can be optimized for each environmental-noise-group-of-country and thus can be accommodated in the headset in addition to cloud. Further, the training dataset can also be simplified due to a limited scope.

Additionally, the total number of purposely-built pre-trained flows should be optimum for better model or flow management purposes. As described above, this step is optional since existing Krisp or BabbleLabs APIs may be used as the purposely-built pre-trained flows. Otherwise, the following algorithm may be followed to create the dataset, which will help to decide the optimum number of specialized purposely-built pre-trained flows:

1. Determine the types of location based on the different intensity of noise levels and noise characteristics.
2. As per environmental noise levels, classify the countries into multiple noise-groups (e.g., high, moderately high, moderate, low, etc.).
3. For every noise-group of countries, create the noise level dataset for different types of locations (see #1 above).
4. For every noise-group of countries:
   A. Create different noise bands based on noise characteristics obtained from the noise level dataset and attach the location types with them. Every noise-group should contain at the most four such noise bands.
   B. Create a training dataset corresponding to every noise band in every noise group by capturing and synthesizing relevant noises for that band (in the context of relevant location types).
   C. Create a specialized AI model for every noise band and train the same on the training data.

5. Thus for all the noise-group of countries at the most there can be approximately 16 to 20 specialized purposely-built pre-trained flows which will be trained on handling varieties of noises in different locations around the world.
6. There can be some commonalities among a few noise bands belonging to various noise-group of countries. With proper exploratory data analysis (EDA) one can further optimize them and thus in turn the total number of specialized purposely-built pre-trained flows will be optimized as well.

The above approach will aid in developing a complete and moreover intelligent and specialized noise suppression or cancellation mechanism that can handle any kind of unwanted noise across the globe.

As described previously, in different countries around the world non-stationary noise intensity levels vary significantly from location to location, whereas current AI-driven headsets are trained on a generic noise related dataset to filter out noise. It is well known that acoustic data is highly nonlinear, not quite smooth, and speech processing is quite complicated from an AI algorithm efficiency perspective. Even after employing manifold learning, the data manifold remains complicated in case of acoustic dataset. Hence, if one focuses on using location specific noise data from across the globe for the most efficient noise handling purpose by a generic model, the model will become quite complicated and may not be that efficient. However, through aspects of the techniques presented herein the specialized purposely-built models will be trained on the proposed noise band of data (which ought to contain a huge dataset of good varieties in that band) which can be managed (e.g., high dimension to low dimension conversion) very easily using manifold learning and the models will definitely outperform the existing generic model or flow.

In summary, techniques have been presented that support an AI based, state of the art, intelligent, and interactive algorithm that will detect current headset location based on a geolocation tag and invoke the appropriate specialized purposely-built pre-trained flows to cancel or suppress noise.